

Overlay Generation Language/370



Overlay Generation Language/370: User's Guide and Reference

Overlay Generation Language/370



Overlay Generation Language/370: User's Guide and Reference

Note

Before using this information and the products it supports, be sure to read the general information in "Notices" on page 391.

Fifth Edition (July 2002)

This edition contains information from and makes obsolete the *IBM Overlay Generation Language/370: User's Guide and Reference*, S544-3702-03. Changes or editions to the text are indicated by vertical bars in the left margin.

This is the fifth edition of *IBM Overlay Generation Language/370: User's Guide and Reference*, S544-3702. This edition applies to the IBM Overlay Generation Language/370 (OGL/370), Release 1.0, Licensed Program (Program Number 5688-191), and to any subsequent release of the program until otherwise indicated in new editions or technical newsletters. OGL/370 Program Number 5688-191 replaces three separate programs: 5665-308 for MVS, 5664-293 for VM, and 5666-324 for VSE.

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS MANUAL "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this may not apply to you.

IBM does not warrant that the contents of this publication or the accompanying source code examples, whether individually or as one or more groups, will meet your requirements or that the publication or the source code is error-free.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer, your IBM Marketing Representative, or your IBM Printing Systems Company Marketing Representative.

The IBM Printing Systems Company welcomes your comments. For your convenience, a form for reader's comments is provided at the back of this publication. If the form has been removed, you may send your comments to the following address:

INFORMATION DEVELOPMENT
IBM PRINTING SYSTEMS
DEPARTMENT H7FE BUILDING 004M
PO BOX 1900
BOULDER CO 80301-9191

If you prefer to send your comments electronically, use one of the following methods:

- Internet: printpub@us.ibm.com
- Fax: 1-800-524-1519

Internet: Visit our home page at <http://www.ibm.com/printers>

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

However, the following copyright notice protects this documentation under the Copyright laws of the United States and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

© Copyright International Business Machines Corporation 1984, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Tables	xiii
Summary of Changes	xv
Fifth Release Changes	xv
Fourth Release Changes	xv
Third Release Changes	xv
First Release Changes	xv

Part 1. Introduction to OGL/370 1

Chapter 1. Introduction to OGL/370	3
What Is an Overlay?	3
Concepts	4
Commands	4
What Programs You Need	16
What You Need to Know	16
About This Manual	17
Major Divisions of This Publication	17
Printers	18
Illustrations	18
Related Publications	19

Part 2. Designing Overlays 21

Chapter 2. Designing a Simple Overlay	23
Conventions	24
Writing Comments	25
Getting Started	26
Specifying Storage, Message, and SOSI Options (CONTROL)	26
Beginning the Overlay (OVERLAY)	27
Orienting the Overlay (ORIENT)	28
Drawing a Grid (DRAWMASK)	33
Drawing Rules	36
Positioning the Rule (POSITION)	37
Defining the Rule (DRAWRULE)	38
Drawing Boxes	41
Positioning the Box (POSITION)	41
Defining the Box (DRAWBOX)	41
Relative Positioning (POSITION)	46
Fine-Tuning the Overlay	49
Cleaning Up	51
Using Defaults and Abbreviations	51
Chapter 3. Drawing Circles and Paths	53
Drawing Circles	53
Positioning the Circle (POSITION)	55
Defining the Circle (DRAWCIRCLE)	55
Drawing Paths	58
Positioning the Path (POSITION)	59
Defining the Path (DRAWPATH)	60

Chapter 4. Adding Text	65
Text Orientation and Positioning	65
Terms Used to Describe Text	65
SOSIINOSOSI Option of the CONTROL Command	65
Specifying Text	66
Parts of a Text Line.	66
Specifying Single-Byte Text	67
Specifying Double-Byte Text with the NOSOSI Option	67
Specifying Double-Byte Text with the SOSI Option	67
Text Margins in Boxes and Circles	70
Text Margins in Boxes.	70
Text Margins in Circles	73
Positioning and Spacing the Text.	76
Text Format	76
Text Orientation	85
Text Placement	86
Ensure the Text Fits	99
Adding Text to Boxes (DRAWBOX WITHTEXT)	100
Adding Text without Boxes.	101
Adding Text to Circles (DRAWCIRCLE WITHTEXT)	106
Specifying the Text for DRAWBOX and DRAWCIRCLE	106
Adding Text Another Way (SETTEXT)	108
Symbolic Data Sets or Files	114
Chapter 5. Adding Graphics	117
Choosing Graphics (SEGMENT)	117
Designing Graphics (DEFINE)	119
Positioning Graphics (POSITION)	125
Placing Graphics (PLACE)	125
Chapter 6. Adding Color	131
Putting it All Together	132
Part 3. Additional Features	135
Chapter 7. Additional Features and Commands	137
Using Default Options (SETUNITS)	137
General Features of SETUNITS	138
Default Units of Measurement	138
SETUNITS Example	139
Line Spacing with SETUNITS	141
Corner Length Values with SETUNITS	141
Text Margins with SETUNITS	144
Top-Left and Center Positioning with SETUNITS	144
SETUNITS and DRAWMASK	147
Repeating Rules (DRAWRULE REPEAT)	149
Repeating Boxes (DRAWBOX REPEAT)	154
Varying Shading and Text	159
Defining and Placing Groups	162
Positioning a Group (POSITION)	162
Defining a Group (DEFINE GROUP)	163
Positioning inside a Group (POSITION)	164
Using SETUNITS inside a Group (SETUNITS)	164
Placing a Group (PLACE)	164
More Features of DRAWBOX	167
Shading with Dotted and Dashed Borders	168

Keywords	169
Chapter 8. Additional Features for Circles and Paths.	171
More Features of DRAWCIRCLE	173
Dotted and Dashed Borders	174
Repeating Circles (DRAWCIRCLE REPEAT)	175
Varying Shading and Text	177
More Features of DRAWPATH	179
Trimmed Path Ends	182
Length of Rounded Connections	183
Repeating Paths (DRAWPATH REPEAT)	186
Varying Shading	187
<hr/>	
Part 4. Reference	189
Chapter 9. Overlay Generation Language Commands	191
Syntax Rules for Commands	191
General Syntax Rules	191
Reading Syntax Diagrams	192
CBDUMP Command	194
CONTROL Command	195
DEFINE BARCODE Command	197
DEFINE COLOR Command	199
DEFINE GROUP Command	201
DEFINE OBJECT Command	202
DEFINE PATTERN Command	204
DRAWBOX Command	206
DRAWBOX COLOR Command	209
DRAWBOX Command-Spaced Repetition	209
DRAWBOX Command-Location Repetition	211
DRAWBOX Command-Shading Boxes	211
DRAWBOX WITHTEXT Command	213
DRAWCIRCLE Command	217
DRAWCIRCLE Command-Spaced Repetition	221
DRAWCIRCLE Command-Location Repetition	223
DRAWCIRCLE Command-Shading Circles	224
DRAWCIRCLE WITHTEXT Command	225
DRAWGRAPHIC Command	229
DRAWGRAPHIC BOX Command	229
DRAWGRAPHIC CIRCLE Command	231
DRAWGRAPHIC ELLIPSE Command	233
DRAWGRAPHIC PATH Command	234
DRAWGRAPHIC FILLETS Command	235
DRAWGRAPHIC MARKER Command	236
Programming Samples	241
DRAWMASK Command	242
DRAWPATH Command	243
DRAWPATH Command - Specifying Path Points	247
DRAWPATH Command-Spaced Repetition	249
DRAWPATH Command-Location Repetition	250
DRAWPATH Command-Shading Paths	251
DRAWPATH Command-Tips	252
DRAWRULE Command	255
DRAWRULE Command-Spaced Repetition	256
DRAWRULE Command-Location Repetition	257
FONT Command (MVS)	258

FONT Command (VM)	260
FONT Command (VSE)	262
ORIENT Command	264
OVERLAY Command	267
PLACE BARCODE Command	269
Examples	273
PLACE Command - Segments and Groups	275
PLACE OBJECT Command	276
PLACE PATTERN Command	278
POSITION Command	280
SEGMENT Command (MVS)	283
SEGMENT Command (VM)	284
SEGMENT Command (VSE)	285
SETTEXT Command	286
SETUNITS Command	291
Appendix A. Sample Overlay RCPT: Output Listing	295
VM Output Listing for Overlay RCPT	295
MVS Output Listing for Overlay RCPT	303
VSE Output Listing for Overlay RCPT	304
Appendix B. Data-Set and File Allocation	305
MVS Data-Set Allocation	305
Sequence numbers	305
Allocation of OVERLIB and SAMPLE data sets	305
VSE File Allocation	306
Sequence Numbers	306
VM File Allocation	306
Sequence Numbers	306
Appendix C. System Dependent Information Procedures	307
In the MVS Environment	307
//STEP1	307
//OUTPUT1	307
//SYSPRINT	308
//SAMPLE.	308
//OVLIB	308
//FONTDD	308
//SYMBOLIC.	308
//SEGDD	308
//SYSIN	308
In the VSE Environment	309
* \$\$ JOB	309
//JOB	309
* \$\$ LST	309
//ASSGN	310
//DLBL	310
//EXTENT.	310
//LIBDEF	310
//EXEC	311
In the VM Environment	312
Using the Native OGL/370 (VM) Command	312
Using OGLVM EXEC.	313

Using a PSF EXEC	313
Appendix D. Merging Overlays and Variable Data	315
Coordinating Form Definitions and Page Definitions	315
Translating Measurements.	318
Appendix E. The Symbolic Data Set and Symbolic File	319
Using the Symbolic Data Set in MVS.	319
Using the Symbolic File in VSE	319
Using the Symbolic File in VM	320
Symbolic Data Sets and Symbolic Files Containing Double-Byte Characters	320
Appendix F. Matching Fonts with Text Formatting	323
Appendix G. Shade Patterns and Types	325
Appendix H. Printer Characteristics	335
Printable Area	335
Text Direction Restrictions	335
Printer Storage Limitations.	335
Appendix I. OGL/370 Keywords	337
Modifying the Command Keywords	338
Keywords and Code Numbers	339
Appendix J. Storage Summary	341
Appendix K. Measurement Units Conversion Table	343
Appendix L. Codes and Messages	345
Codes	345
Messages.	345
Notices	391
Programming Interfaces	393
Trademarks	393
EuroReady	394
Glossary	395
Glossary	395
Index	403

Figures

1. Sample Overlay	3
2. Physical Page and Logical Page for Continuous Forms and Cut-Sheet Printers	7
3. The Same Medium Overlay Printed on Multiple Pages	8
4. The Same Page Overlay Printed on Multiple Pages	9
5. Coordinates and Origin	10
6. Printing with Pels	11
7. Rule Thickness	11
8. Orientation	12
9. Relative Orientation	13
10. Character Boxes	13
11. Uniformly-Spaced and Typographic Fonts	14
12. Fonts Used in Overlay RCPT	14
13. Orientation and Format	15
14. Overlay RCPT	23
15. Command Sample	24
16. CONTROL Command	26
17. OVERLAY Command	27
18. Overlay Specified with 0° Orientation for the IBM 3800	29
19. Medium Overlay Specified with 90° Orientation	30
20. Page Overlay Specified with 90° Orientation	31
21. ORIENT Command	31
22. OFFSET Subcommand and Orientation	32
23. DRAWMASK Command	33
24. Spacing Intervals for a Grid	34
25. The Mask for Overlay RCPT	34
26. Rules	36
27. POSITION Command	37
28. DRAWRULE Command	38
29. Rule Thicknesses	38
30. Rule Types for a 4-Pel Rule	39
31. DRAWRULE Commands for Overlay RCPT	40
32. Boxes	41
33. POSITION Command	41
34. DRAWBOX Command	42
35. Box Size	42
36. Exact Box Placement	43
37. Named Shade Types for Boxes (STANDARD)	44
38. DRAWBOX Commands for Overlay RCPT	45
39. Printed Rules and Boxes	45
40. POSITION Command	46
41. The Initials with Distances	47
42. A New Way to Use the POSITION Command	48
43. Fine-Tuning the Overlay	49
44. Extract from Overlay “RESULT” Used to Illustrate Circles	54
45. POSITION (Circle) Command	55
46. DRAWCIRCLE Command	55
47. Circle Size	56
48. Half-Circles	57
49. Quarter-Circles	57
50. Extract from “RESULT” Overlay Used to Illustrate Paths	58
51. Positioning of a Rule and a Path	59
52. POSITION (Path) Command	59
53. DRAWPATH Command	60

54. Path Connection Types	61
55. DRAWPATH Command	62
56. LINE Subcommand for Single-Byte Text	66
57. Examples of Text Segments and Text Strings in the LINE subcommand	66
58. Example of LINE Subcommand for Single-Byte and Double-Byte Text with NOSOSI	67
59. Example Using Double-Byte Characters and the SOSI Option	68
60. Example Using Double-Byte Characters and the SOSI Option	69
61. Example Using Double-Byte HEX Characters and the SOSI Option	70
62. DRAWBOX Text Margins for a Box with All Corners Rounded	71
63. DRAWBOX Text Margins for a Box with One Rounded Corner	72
64. DRAWBOX Text Margins for a Box with Two Adjacent Rounded Corners	72
65. DRAWBOX Text Margins for a Box with Two Opposite Rounded Corners	73
66. DRAWBOX Text Margins for a Box with Three Rounded Corners	73
67. Text Margins in a WHOLE Circle	74
68. Text Margins in a $\frac{3}{4}$ Circle.	74
69. Text Margins in a $\frac{1}{2}$ Circle.	75
70. Text Margins in a $\frac{1}{4}$ Circle.	75
71. Text Positioning Options in a Box.	77
72. Text Positioning Options in a Circle — ROUNDED Margin.	77
73. Text Positioning Options in a Circle — SQUARE Margin	78
74. Placement of Text Written in MODERN Format.	79
75. Placement of Text Written in MODERN Format in a SQUARE Text Margin.	80
76. Placement of Text Written in COLUMN Format.	81
77. Placement of Text Written in COLUMN Format in a SQUARE Text Margin.	82
78. Placement of Text Written in TATE Format	83
79. Placement of Text Written in TATE Format in a SQUARE Text Margin	84
80. Text Block Orientations for a Circle	85
81. Text Block Orientations for a Box	86
82. Justified Text in a Box	87
83. Justified Text in a Circle	88
84. Placement of Balanced and Justified Text, Written in MODERN Format.	89
85. Placement of Balanced and Justified Text Written in MODERN Format in a SQUARE Text Margin	90
86. Placement of Balanced and Justified Text Written in COLUMN Format	92
87. Placement of Balanced and Justified Text Written in TATE Format.	93
88. Placement of Balanced and Justified Text Written in COLUMN Format in a SQUARE Text Margin	94
89. Placement of Balanced and Justified Text Written in TATE Format in a SQUARE Text Margin	95
90. Balanced Text With Blanks	96
91. WITHTEXT Command.	97
92. Line Spacing in a Box	98
93. Line Spacing in a Circle	98
94. Text Exceeding Boundaries	99
95. Box with Text.	100
96. WITHTEXT Subcommand	101
97. An Invisible Box with Text	102
98. Text Placement Within an Invisible Box Version 1	102
99. Text Placement Within an Invisible Box Version 2	102
100. Printed Text	103
101. Sample Overlay without Graphics	105
102. Circle with Text	106
103. Text Origin with SETTEXT	108
104. SETTEXT	109
105. Alignment of Text Written with MODERN	110
106. Alignment of Text Written with COLUMN	110
107. Alignment of Text Written with COLUMN	111
108. Text Printed Using SETTEXT	113
109. Symbolic Data Set or File Records	114

110. SEGMENT Command for MVS	117
111. Two Ways of Looking at the Same Image	119
112. Overlay Mask for Designing Graphics.	120
113. DEFINE Command	120
114. Pel Pattern for the “T”	122
115. Pel Pattern for the “C”	123
116. The DEFINE Command for the “C” Pattern.	124
117. Invisible Boxes around Graphics.	125
118. PLACE Segment	125
119. PLACE Pattern	125
120. Pattern Orientation.	126
121. Pattern Shading.	127
122. MIRROR Option	127
123. NEGATIVE Option	128
124. Commands for Drawing Graphics (MVS)	128
125. Graphic Origins	129
126. Graphics in Different Orientations and the Same Positions	129
127. SETUNITS Command	139
128. Choosing the Line-Spacing Default.	141
129. Length of a Rounded Box Corner	142
130. Corner Length of a Rounded Path Connection	142
131. Boxes with SMALL , MEDIUM , and LARGE Corner Length	143
132. HALF and MAX Corner Length for Boxes	144
133. HALF and MAX Corner Length for Paths	144
134. TOPLEFT and CENTER Positioning for a Box	145
135. TOPLEFT and CENTER Positioning for a Rule	146
136. Sample Overlay PREREG	147
137. Repeated Rules.	149
138. Spaced and Location Rule Repetition.	150
139. DRAWRULE REPEAT (Spaced Repetition)	150
140. Spaced Repetition for DRAWRULE	151
141. DRAWRULE REPEAT	153
142. Repeated Boxes	155
143. Spaced and Location Box Repetition	155
144. DRAWBOX REPEAT	156
145. Spaced Boxes	157
146. Boxes Spaced 0	157
147. DRAWBOX REPEAT (Location Repetition).	158
148. Numbering Boxes, Created with Location Repetition	159
149. Numbering Boxes, Created with Spaced Repetition.	160
150. DRAWBOX REPEAT (Shading)	160
151. DRAWBOX REPEAT (WITHTEXT)	161
152. A Group	162
153. DEFINE GROUP Commands.	163
154. Box with Rounded Corners and Diagonals	167
155. DRAWBOX Subcommands	167
156. Shading Effects in Boxes	169
157. Overlay RESULT	172
158. Sample Circle	173
159. DRAWCIRCLE Subcommands	173
160. The effects of the AXIS subcommand.	174
161. Repeated Circles	175
162. DRAWCIRCLE REPEAT (Spaced Repetition)	175
163. DRAWCIRCLE REPEAT (Location Repetition)	176
164. DRAWCIRCLE REPEAT (Shading)	177
165. Extract from Overlay RESULT Used to Illustrate Paths	179

166.	DRAWPATH Command	180
167.	Open Path with Rounded Ends	181
168.	PATHEND Subcommand	181
169.	Path End Shapes	182
170.	Examples of Path End Shapes in a Box	182
171.	Examples of TOPLEFT and CENTER Positioning	183
172.	Two Corner Lengths	184
173.	DRAWPATH Command with Corner Lengths	184
174.	DRAWPATH Command with Corner Lengths	185
175.	Repeated Paths.	186
176.	How to Measure the Width and Height of a Box	208
177.	Spaced Boxes	210
178.	Boxes with 0 Spacing	210
179.	How to Measure the Radius of a Circle	219
180.	Effects produced using the AXIS (AXES) subcommand	220
181.	Spaced Circles	222
182.	Circles with DIAMETER Spacing	222
183.	Ellipse parameters	237
184.	Fillets	238
185.	Fill Patterns	240
186.	Path End Shapes	245
187.	Path Connection Types	246
188.	Examples of MITER Connections	252
189.	Examples of ROUNDED Connections.	253
190.	Examples of Closed and Disjointed Paths	254
191.	Overlay Offset and Overlay Origin for ORIENT 0 Overlay	265
192.	Overlay Offset and Overlay Origin for ORIENT 90 Overlay	265
193.	Overlay Offset and Overlay Origin for ORIENT 180 Overlay	266
194.	Overlay Offset and Overlay Origin for ORIENT 270 Overlay	266
I 195.	Examples for Placing Barcodes	273
I 196.	Examples for Placing Two-dimensional Barcodes (1 of 2)	273
I 197.	Examples for Placing Two-dimensional Barcodes (2 of 2)	274
198.	Examples for Placing Objects through AFP Workbench	277
199.	Examples for Placing Objects through PSF/MVS.	277
200.	Named Shades for a Pattern	279
201.	Alignment Options Using the MODERN Format	287
202.	Alignment Options Using the COLUMN Format	288
203.	Alignment Options Using the TATE Format.	288
204.	Choosing the Line-Spacing Default.	292
205.	TOPLEFT and CENTER Positioning for Boxes	294
206.	Sample JCL for OGL/370 (MVS)	307
207.	Sample JCS for OGL/370 (VSE).	309
208.	Submitting an Overlay for Compilation	313
209.	Variable and Overlay Text	316
210.	Font Prefixes for Orientation and Format Combinations	324
211.	Shade Pattern — STANDARD	326
212.	Shade Pattern — SCREEN	330

Tables

1.	OGI/370 commands for Graphics in Different Orientations	130
2.	Spacing Value	256
3.	Data-Set Attributes (MVS)	305
4.	File Attributes (VSE)	306
5.	File Attributes (VM)	306
6.	Keywords and Code Numbers	339

Summary of Changes

Fifth Release Changes

This release adds the following:

- Changes in printer terminology.
- The addition of the following colors which may be defined with the **DEFINE COLOR** command:
 - DARKBLUE
 - ORANGE
 - PURPLE
 - DARKGREEN
 - DARKCYAN
 - MUSTARD
 - GRAY

See “**DEFINE COLOR** Command” on page 199 for more information.

- Changes to the **DEFINE OBJECT** command to support new features of the Infoprint Color 130 Plus printer.
- The capability to define and place barcodes in your overlay with the following commands:
 - **DEFINE BARCODE**
 - **PLACE BARCODE**

See “**DEFINE BARCODE** Command” on page 197 and “**PLACE BARCODE** Command” on page 269 for more information.

Fourth Release Changes

This release reformats the command reference chapter and eliminates the separate die cut tabs.

Third Release Changes

This release includes minor technical and usability changes.

Chapter 4, “Adding Text” on page 65 has been rewritten for better understanding and usability.

Two new subcommands, **SOSI** and **NOSOSI**, have been added to the **CONTROL** command.

Any errors found in figures and syntax diagrams have been corrected.

Appendix G, “Shade Patterns and Types” on page 325 contains shading samples and the source code used to print them.

First Release Changes

Prior to OGL/370, there were three OGL products, one for each of the operating systems (MVS, VM, and VSE). The first release incorporated changes for the single product (OGL/370), which operates with the three operating systems.

- The system-specific details, such as system invocation, for each operating system.
- The two new subcommands:
 1. **AXIS (AXES)** subcommand for **DRAWCIRCLE**, which allows you to specify the arrangement of dots and dashes around circles.
 2. **TRACEALL** subcommand for **CBDUMP**, which allows you to specify the level of tracing.
- The distribution of dots and dashes along boxes, paths, and circles.
- Variable block size for MVS.

Part 1. Introduction to OGL/370

An overlay can be used in a printing job in three ways:

- Paper, preprinted with an overlay, is loaded into the printer. Variable information is then added.
- A negative picture of the overlay is loaded into the printer and “flashed” onto the paper. This is a feature of the IBM 3800 Printing Subsystem.
- The overlay is in the form of coded instructions, which is stored in an area called a *library*. The overlay is called for whenever it is needed. This last type of overlay, called an *electronic overlay*, is usually the most convenient. It is easily modified and it does not require anyone to load different paper or to change negatives.

OGL/370 is a tool used for producing electronic overlays. To use this tool to print overlays, you need an IBM licensed program called Print Services Facility (PSF) and an IBM advanced-function printer.

PSF puts the instructions for the overlay, which were generated by OGL/370, into a form that the printer can use. PSF also merges variable data with the overlay. For the sample overlay, shown in Figure 1 on page 3 the variable data includes the student’s name, social security number, and address.

OGL/370 can produce overlays with:

- Horizontal and vertical rules of various thicknesses, colors, and types
- Boxes with different shades and patterns, colors, border thicknesses, and border types
- Circles and partial circles with different shades and patterns, colors, border thicknesses, and border types
- Arbitrary shapes, called paths, with different shades and patterns, border thicknesses, and border types
- Text in various positions with different font styles
- Straight lines connecting any two points
- Graphic images (raster images either already created and stored in a library, or defined in the overlay)

Concepts

Before you learn how to format overlays, you should know the major concepts of OGL/370:

- Commands
- Defaults
- Syntax
- Comments
- Messages
- Medium and page overlays
- Coordinates and Origin
- Pels
- Rules and Borders
- Orientation
- Fonts
- Text
- Graphics
- Groups

Commands

OGL/370 consists of seventeen commands. The following command specifies the size and placement on the paper, for the overlay named “RCPT”, shown in Figure 1.

```
OVERLAY RCPT SIZE 7.25 IN 3.25 IN OFFSET 0 .5 IN;
```

The **OVERLAY** command contains two *subcommands*, **SIZE** and **OFFSET**, each consisting of two *entries*:

- SIZE** This subcommand has entries for width and height: 7.25 inches and 3.25 inches, respectively.
- OFFSET** This subcommand has one entry for each coordinate of the overlay (coordinates are explained later in this chapter).

The group of commands for a particular overlay is called the *overlay definition*. Most commands can occur many times in a single definition, but the following commands can occur only once in a single definition:

CONTROL
OVERLAY
ORIENT

Defaults

You can enter each option in a command, or let OGL/370 make some choices for you. If you let OGL/370 choose, it uses automatic choices called *defaults*. For example, you can define **SOLID**, **DASHED**, or **DOTTED** rules. If rule type is not specified, OGL/370 defaults to **SOLID**.

Note: Defaults are indicated by **UNDERLINED BOLD UPPERCASE** type throughout this book.

If an option has a default (for example, the rule type), do one of three things:

- Enter the default (in this case, **SOLID**).
- Let OGL/370 choose for you (do not enter any command).
- Make a choice other than the default (enter **DASHED** or **DOTTED**).

If you are not familiar with the use of defaults, do not use them while you are learning OGL/370. The defaults are described in detail in “Using Defaults and Abbreviations” on page 51.

Syntax

Syntax, the rules for writing commands, is very exact; if you spell a word differently, leave out a word, or write a word in the wrong order, the command might not do what you want it to do.

General Rules for OGL/370 Commands

Writing commands: The following rules apply to all commands:

- Write only one command per line. However, a single command can extend over more than one line.
- When your command spans more than one line, make sure there is a space in column one of the next line.
- Command keywords, subcommand keywords, and values cannot be split over two lines.
- You can indent lines to make the overlay definition easier to read.
- Leave at least one space between entries of a command. No space is necessary between the last entry in a command and the semicolon (;) that ends the command.
- Commands and subcommands can be entered in uppercase or lowercase, or a combination of the two. For example, OGL/370 sees the following entries in the same way:

```
OVERLAY
overlay
Overlay
```

Note: In this book, commands and subcommands are displayed in uppercase for consistency purposes only.

Keywords: A keyword is part of a command that must be spelled exactly as it appears in this book. Do not use keywords as names (values). A complete list of keywords appears in Appendix I, “OGL/370 Keywords” on page 337.

Sequence: Enter the parts of a command in the order in which they are presented in the examples and diagrams.

Numbers: Specify no more than four digits to the left of a decimal point and no more than two digits to the right. Thus, the largest number that can be specified is 9999.99 and the smallest number (apart from 0) is 0.01.

Orientation

Specify only 0, 90, 180, and 270 degrees. Orientation is explained later in this chapter.

Repetitions, rule thicknesses, and pel

Specify only whole numbers (no decimal points) up to four digits.

End marker: End every command with a semicolon (;).

Comments

OGL/370 lets you explain in the overlay definition what a command is doing. Explanations, called comments, have no effect on the program itself. The use of comments is described in “Writing Comments” on page 25.

Messages

OGL/370 sends messages under certain conditions. When OGL/370 processes an overlay, you can receive a source listing (that is, a printout of the overlay definition and any messages that may result). Source listings for the “RCPT” sample overlay presented in this book is located in Appendix A, “Sample Overlay RCPT: Output Listing” on page 295.

There are three types of messages:

- | | |
|----------------|--|
| All | All information is presented about the command. |
| Warning | There is an error that probably results in undesirable output. |
| Error | There is a definite error, and the overlay may not print as you defined. |

Refer to Appendix L, “Codes and Messages” on page 345 for the text of each message.

You might also receive messages from PSF when the overlay is sent to the printer. See *Print Services Facility Messages* for your operating system for an explanation of these messages.

Medium and Page Overlays

OGL/370 creates overlays that can be used in two ways:

Medium Overlays: Can be placed on a physical sheet of paper.

Page Overlays: Can be placed selectively in various locations on the logical page.

To understand the difference, you need to be able to distinguish between the *physical page* and the *logical page*:

Physical Page: The actual sheet of paper or other medium (such as a sheet of blank labels) that moves through the printer.

Logical Page: The area defined as the space on the physical page where data can be printed.

Figure 2 on page 7 illustrates a physical page and a logical page for continuous forms and cut-sheet printers.

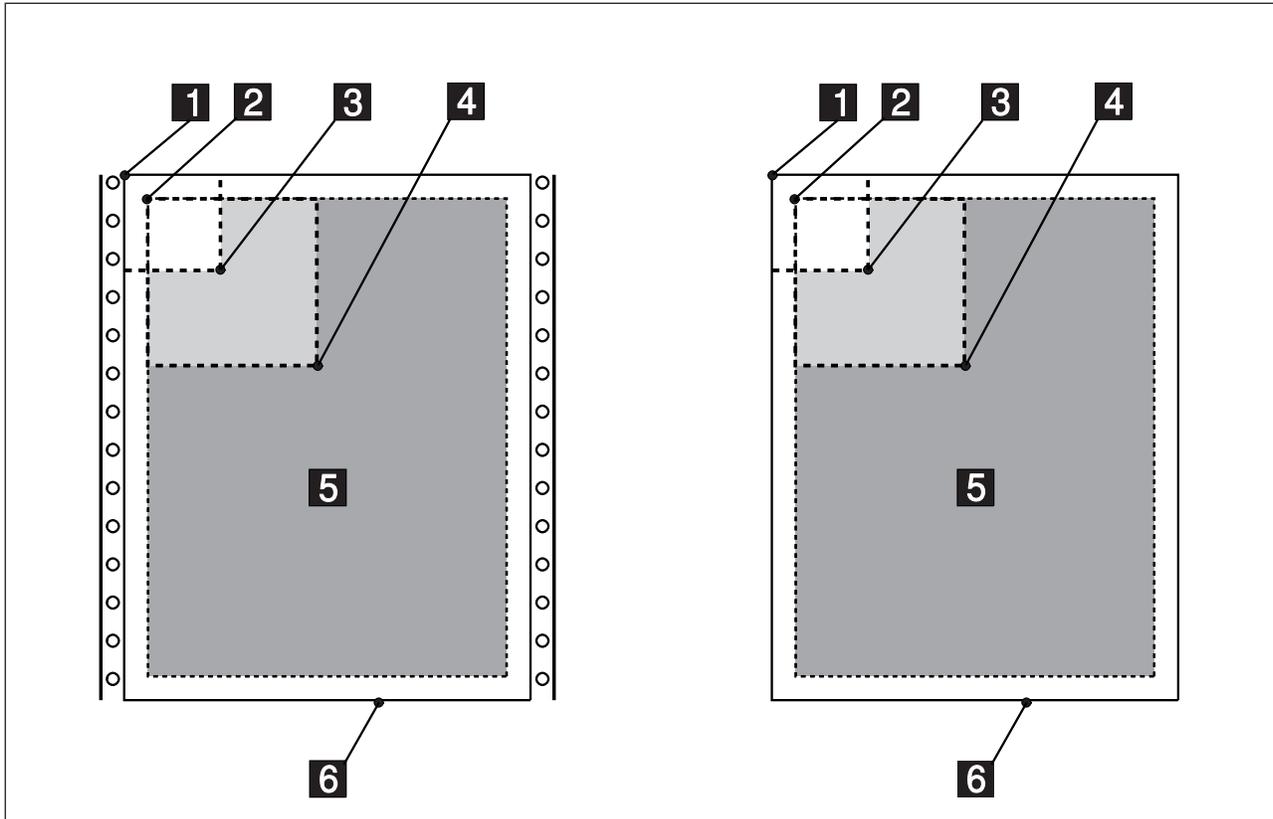


Figure 2. Physical Page and Logical Page for Continuous Forms and Cut-Sheet Printers

- 1** Medium origin
- 2** Logical Page origin specified in a form definition
- 3** Overlay offset specified with the **OFFSET** subcommand on the **OVERLAY** command
- 4** Page overlay origin specified in the Include Page Overlay (IPO) structured field
This offset is added to the overlay offset.
- 5** Logical Page
- 6** Physical Page

An overlay that is positioned in relation to the origin of the physical page is called a *medium overlay*. A medium overlay is positioned like a page of composed text. For example, if the offset of a particular overlay is 1 inch by 1 inch, the top-left corner of the overlay would be placed 1 inch across and 1 inch down from the medium origin.

Some printers have an unprintable area or an area of degraded print quality around the edges of the paper. In order to avoid this area when printing, an offset value must be specified to OGL/370 with the **OFFSET** subcommand on the **OVERLAY** command. For more information on printable areas, see *Advanced Function Printing: Printer Information*

To print a medium overlay, you need to:

- Create the overlay with an offset from the medium origin.
- Name the overlay in a form definition.

The medium overlay that you name in a form definition always prints at the same position on each page as shown in Figure 3. For more information on form definitions, see *IBM Page Printer Formatting Aid/370: User's Guide and Reference*.

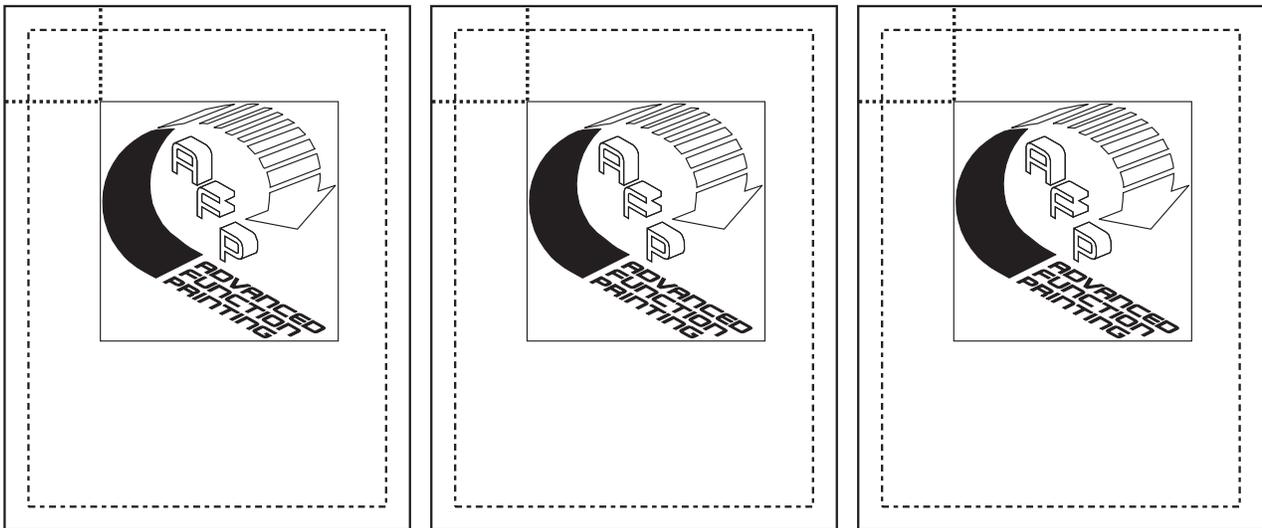


Figure 3. The Same Medium Overlay Printed on Multiple Pages. The AFP overlay was defined with an offset from the medium origin and named in a form definition.

An overlay that is positioned anywhere on the logical page is called a *page overlay*. For example, a page overlay can be positioned as shown in Figure 4 on page 9.

To print a page overlay, you need to:

- Create the overlay with an offset of 0 across and 0 down.
- For line data, name the overlay in a page definition.
- For page data, name the overlay in an Active Environment Group (AEG) structured field.
- Specify the coordinates where you want the overlay printed in the Include Page Overlay (IPO) structured field. The IPO structured field gives you the ability to place a page overlay anywhere on the logical page.

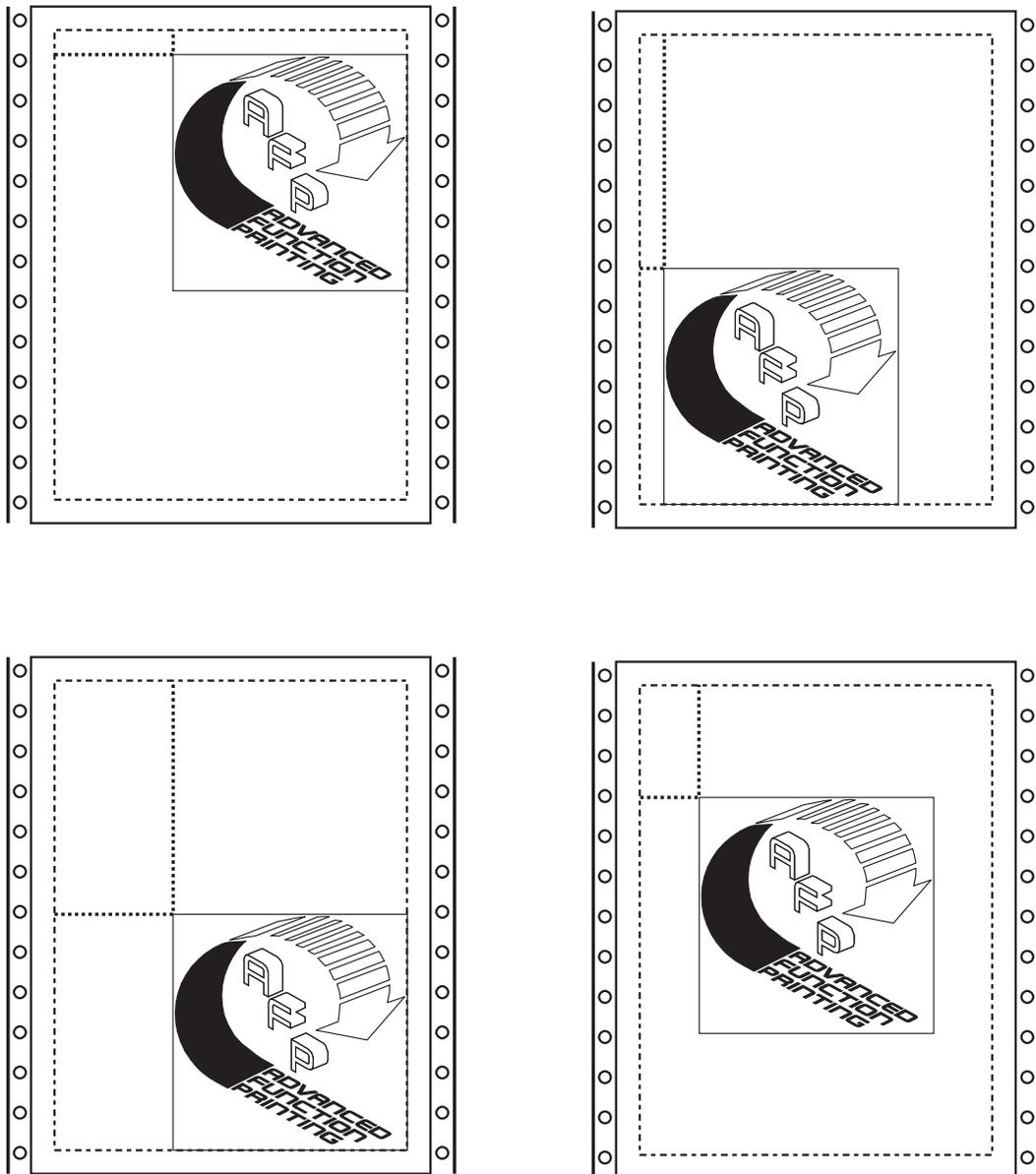


Figure 4. The Same Page Overlay Printed on Multiple Pages. The AFP overlay was defined with a 0,0 offset from the overlay origin, and positioned with the Include Page Overlay structured field.

For more information on page definitions, see *IBM Page Printer Formatting Aid/370: User's Guide and Reference*. For more information on the AEG structured field and on the IPO structured field, see *Advanced Function Printing Data Stream Reference*.

Contact your system programmer for instructions on printing overlays on specific printers.

Note: A medium overlay can be used as a page overlay, if you compensate for the offset value specified with the **OVERLAY** command and the **OFFSET** subcommand.

Coordinates and Origin

An overlay and the paper on which it is printed have two measurements:

- Horizontal
- Vertical

The easiest way to tell someone where to write a letter on a page is to say how far to move horizontally and vertically from a known point. OGL/370 is used to tell the printer the same things. The known point is the *origin*, and the distances are the *coordinates*. The origin is the top-left corner of a page or an overlay. Coordinates and origin are illustrated in Figure 5.

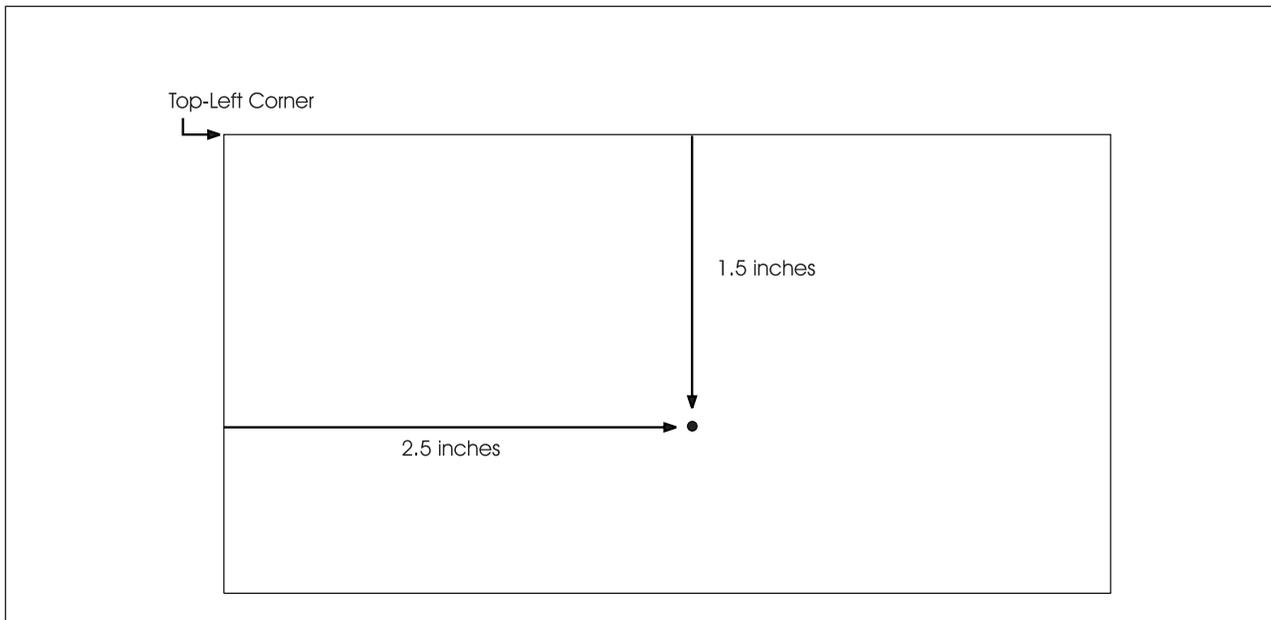


Figure 5. Coordinates and Origin

In Figure 5, the dot is 2.5 inches horizontally and 1.5 inches vertically from the origin. Thus, the coordinates of the dot are 2.5 inches and 1.5 inches.

Picture Elements (Pels)

To understand how rules, graphics, and text are printed, think of the paper as a grid (as in Figure 6 on page 11). Each of these squares represents a pel. The printer can tone some pels and leave others untoned. The patterns of toned and untoned pels create the images, text, rules, and shadings of an overlay.

The size of a pel may vary depending on the type of printer. In this book, all commands and examples are written for a printer that prints 240 pels per inch. Therefore, one pel equals 1/240 of an inch.

Figure 6 on page 11 represents a vertical rule, 2 pels thick and 21 pels long. Its position is also measured in pels, it is 8 pels from the left margin and 5 pels from the top margin.

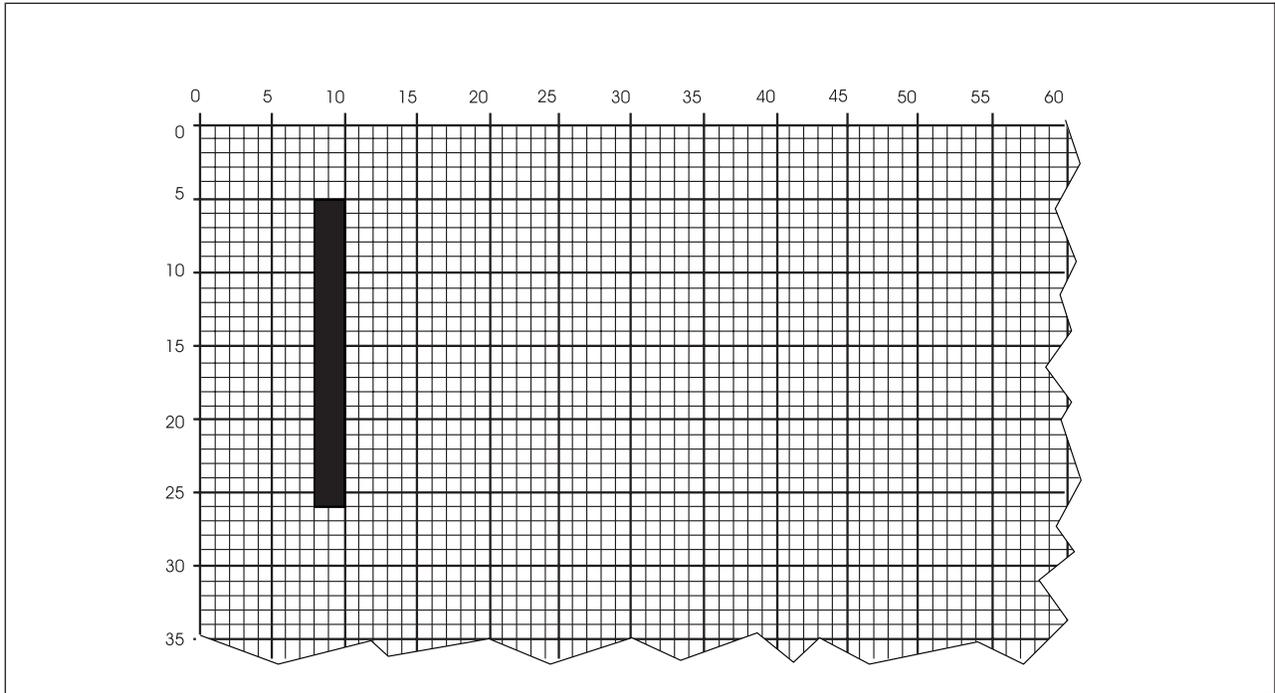


Figure 6. Printing with Pels. (Not actual size.)

Rules and Borders

An important concept to remember is that a printed rule has thickness. For example, OGL/370 allows you to define a rule and then repeat it at spaced intervals. If you specified 1/2-inch intervals, the space between the rules is 1/2 inch (120 pels) minus the thickness of the rule. In Figure 7, for example, the rules are 4 pels thick, the distance between rules is 116 pels (120 minus 4), and the distance from the left side of one rule to the right side of the next is 124 pels (120 plus 4).

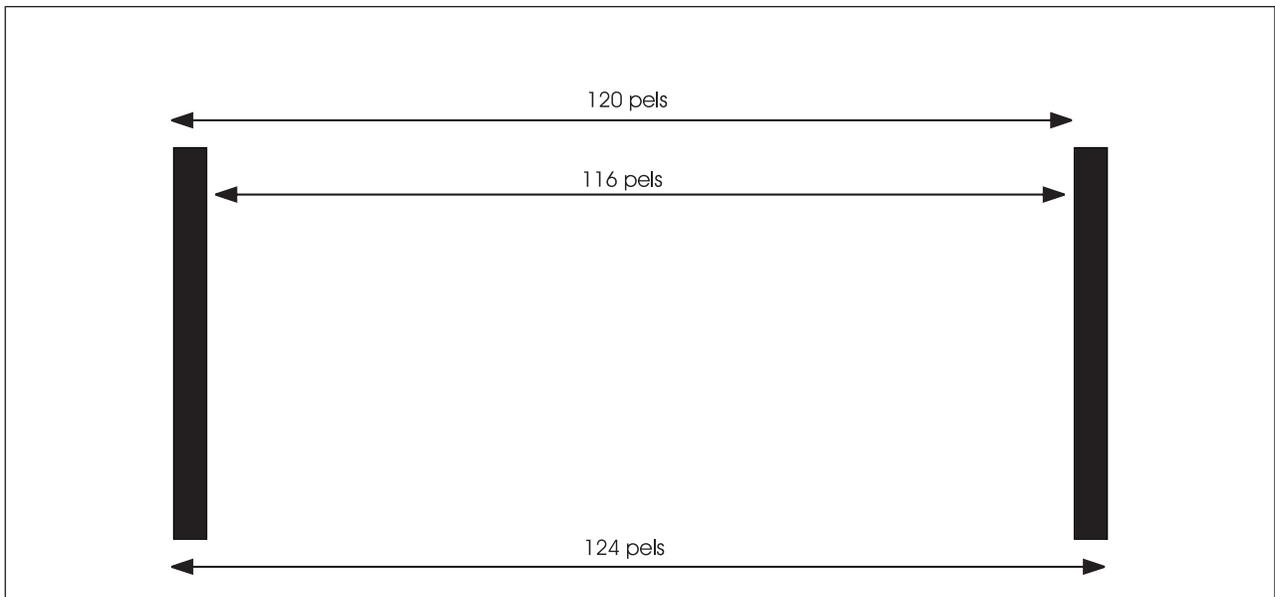


Figure 7. Rule Thickness. (Not actual size.)

For many applications, 4 pels make little or no practical difference. However, ignoring rule thickness can sometimes cause problems. Details of rule thickness and how to avoid problems are described in the sections on the commands for drawing rules, boxes, circles, and paths.

Orientation

The concept of orientation applies to many things on an overlay. For example, the text you are reading right now is in the 0° orientation relative to this page. See Figure 8 for a graphic explanation of orientation.

Note: In Figure 8, the boxes represent overlays.

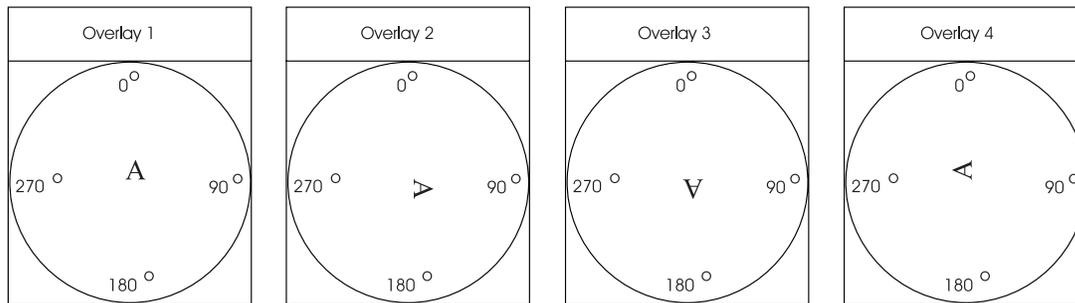


Figure 8. Orientation

- Overlay 1** The top of the “A” points to the “ 0° ”. The letter is in the 0° orientation.
- Overlay 2** The “A” has been rotated clockwise and now points to the “ 90° ”. The letter is in the 90° orientation.
- Overlay 3** The “A” has been rotated to the “ 180° ”. The letter is in the 180° orientation.
- Overlay 4** The “A” points to the “ 270° ”. The letter is in the 270° orientation.

It is important to understand that orientation is relative. For example, in Figure 9 on page 13 we have rotated Overlay 4 and its contents.

Looking at Figure 9 on page 13:

Q: Relative to this page, what is the orientation of the overlay?

A: The overlay is oriented 90° relative to the page.

Q: What is the orientation of the “A” relative to the overlay?

A: The letter is oriented 270° relative to the overlay.

Q: What is the orientation of the “A” relative to the page?

A: The “A” is oriented 0° relative to the page.

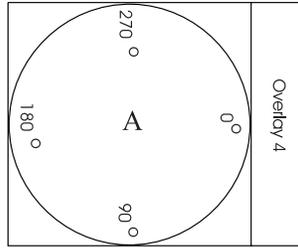


Figure 9. Relative Orientation

You can orient the entire overlay, words, and some types of graphics to any of four orientations (0°, 90°, 180°, or 270°).

See “Graphics” on page 15 for a description of orienting graphics. However, your printer may not be able to print in all of these orientations. These restrictions are described in Chapter 2, “Designing a Simple Overlay” on page 23.

Fonts

You might want to print different blocks of text using different types of characters. A set of characters of a specific size and shape is called a *font*. You specify a certain font by naming it in a **FONT** command and specifying that name in the **LINE** subcommand. When you are designing an overlay, you need to know the size and appearance of the fonts and their official names, called *member IDs*. You have several IBM fonts available, and your company may also have its own fonts. For information about the fonts available in your company, see your system programmer.

OGL/370 sees the characters in a font as rectangles, called *character boxes*. An entire character fits into its character box, usually with room to spare. See Figure 10 for an illustration of character boxes.

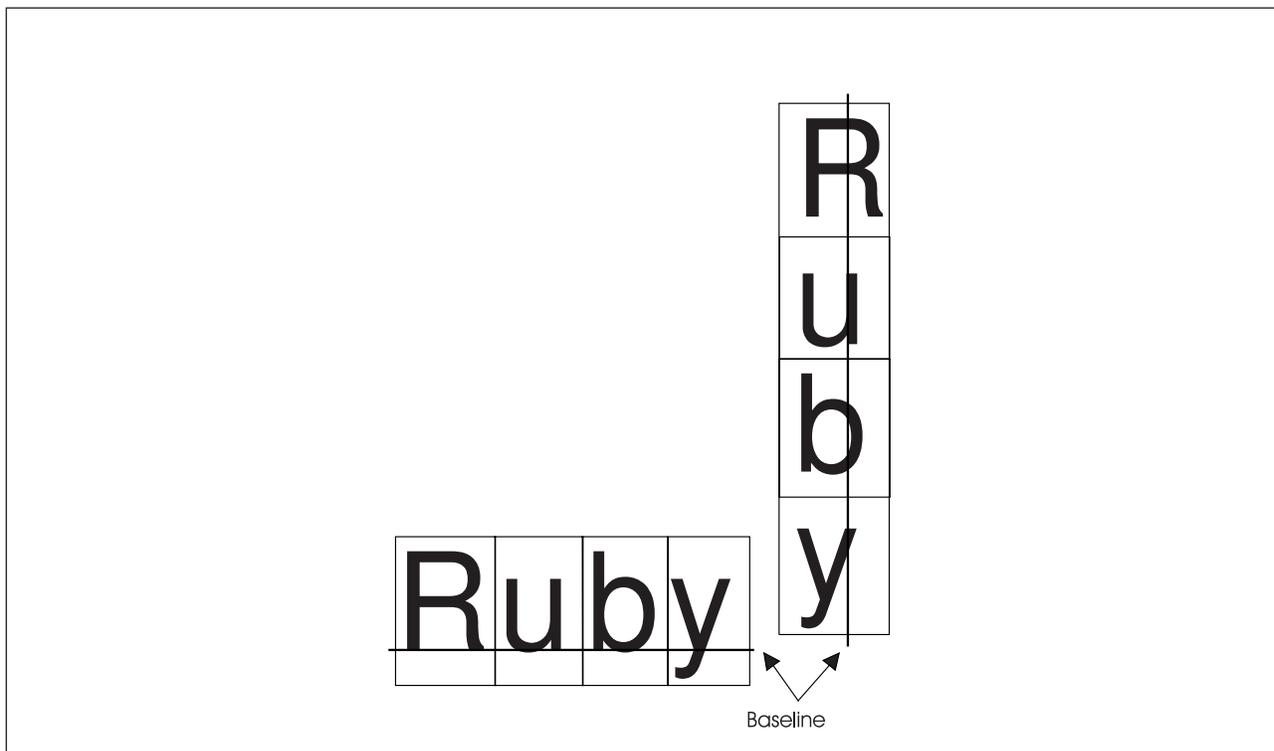


Figure 10. Character Boxes

The *baselines* in Figure 10 on page 13 are invisible rules associated with each font that OGL/370 uses when it places text, including text printed with multiple fonts. Subsequent chapters describe line spacing for text being measured from the baseline of one line to the baseline of the next.

Fonts are divided into two major groups:

Single-byte fonts: Represent phonetic characters, such as the Latin alphabet or the Japanese Katakana alphabet.

Double-byte fonts: Represent nonphonetic characters, such as those in the Kanji writing system used in Japan.

Note: The differences in the use of the two types of fonts are explained later in this book.

Single-byte fonts are further divided:

Uniformly spaced fonts: In a uniformly spaced font, all of the character boxes have the same width. For example, a lowercase “i” and an uppercase “M” have character boxes of the same width. See Part A in Figure 11.

Typographic fonts: In a typographic font, the character boxes can have different widths. See Part B in Figure 11.

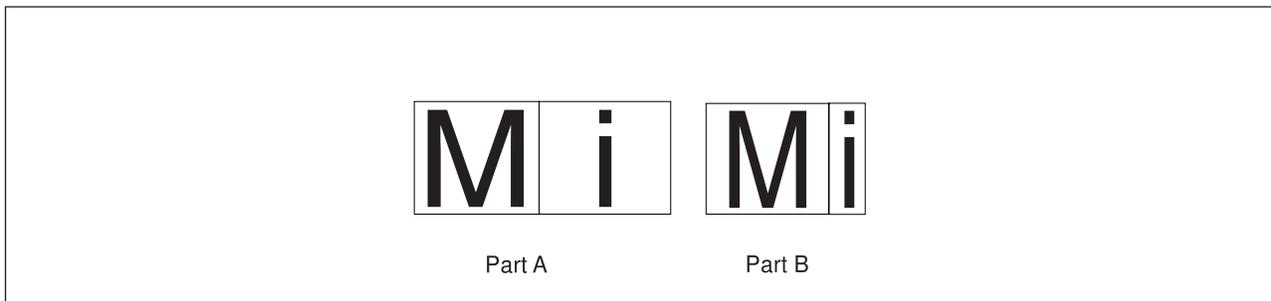


Figure 11. Uniformly-Spaced and Typographic Fonts

Overlay “RCPT” uses five different fonts. The font names and examples are shown in Figure 12.



Figure 12. Fonts Used in Overlay RCPT

Text Format

When you want to print text, you must specify not only the fonts and the orientation but also the format of the text. The format of the text controls the direction your eyes move when you read the text. OGL/370 has three formats:

MODERN In standard format, including English, characters are read from left to right and from top to bottom (default).

COLUMN On some signs and forms, you read the characters from top to bottom and the lines from left to right.

TATE In some Oriental languages, including Japanese, characters are read from top to bottom and lines from right to left.

Orientation: If a certain font is available in the four orientations already mentioned, text written in any of these three formats can be rotated to any of the four orientations. The sample texts in Part One of Figure 13 are oriented 0° relative to the page. If you rotate the text 90°, 180°, or 270°, it is positioned as shown in Parts Two, Three, and Four, respectively.

	Part One (0°)	Part Two (90°)	Part Three (180°)	Part Four (270°)
MODERN	Two lines of text	Two lines of text	Two lines of text	Two lines of text
COLUMN	T o w f o l e i x n t e s	S e u t l e i o f w t o	s e t x n t e l t o f w o t	T o f w o t l e i x n t e s
TATE	o T f w o t e l x i t n e s	S e u t l e i o f w t o	s e t x n t e l t o f w o t	T o f w o t l e i x n t e s

Figure 13. Orientation and Format

Note: Your printer may not print certain combinations of orientation and format. For more details, see Appendix H, "Printer Characteristics" on page 335.

Graphics

Your company might have a collection of stored graphics, called *page segments*, that can be used by naming one or more of them in your overlay definition. A page segment can also include text. To find out about the segments available to you, ask your system programmer.

Using OGL/370, you can also define your own graphics, called having *patterns*, by the printer tone some pels while leaving other pels untone.

Groups

An overlay might include a group of rules, circles, boxes, paths, or text occurring in more than one place on the form. An additional feature of OGL, the **DEFINE GROUP** command, allows you to define that group as a single unit, give it a name, and then call for it by name whenever you need it within that overlay definition.

The details of this command are presented in Chapter 7, “Additional Features and Commands” on page 137.

What Programs You Need

The *OGL/370: Licensed Program Specifications*, (G544-3697), provides the definitive statement of the hardware and software requirements of OGL.

To use OGL/370, you must have one of the following operating systems installed:

- VM/SP 5.0 or later

Note: If you need double byte messages returned to the screen, VM/SP HPO 5.0 (or later) or VM/XA SP 2.0 (or later) is required.

- MVS/SP 1.3.5 or later
- MVS/SP 2.2.0 or later
- MVS/SP 3.1.0 or later
- VSE/SP 3.2.0 or later
- VSE/ESA 1.1.0 or later

To print the sample overlays generated by OGL/370, you must have Print Services Facility (PSF) installed on one of the above operating systems.

Note: The sample overlay generated by OGL/370 is the same regardless of the operating system on which OGL/370 is running. Therefore, the sample overlay may be created on one system, using OGL/370, and printed on another, using PSF.

If you wish to merge other data with your overlays, you will need either the Print Management Facility (PMF) licensed program or the Page Printer Formatting Aid (PPFA) licensed program. PMF and PPFA produce sets of instructions called page definitions and form definitions, which are stored in libraries. These instructions tell PSF the name of the overlay to be used and, if additional data is to be printed on the overlay, where that data should be placed. PMF and the Graphical Data Display Manager (GDDM) licensed programs can generate a type of graphic called a *page segment*. Page segments and fonts are also stored in libraries. This book describes how these library members, fonts, and segments can be used on overlays.

What You Need to Know

You do not need to be a computer programmer to use OGL/370, but you should have a knowledge of:

- The form needs of your company
- The use of a display terminal
- The use of an on-line text editor
- The fonts available in your installation

You may need the help of a programmer to run OGL/370 under your operating system.

About This Manual

This book explains how to use OGL/370 to create electronic overlays for printing forms on IBM AFP printers.

Note: If you are using OGL/370 for the first time or you use OGL/370 occasionally, you should read *OGL/370: Getting Started* along with Chapter 1, “Introduction to OGL/370” on page 3 and Chapter 2, “Designing a Simple Overlay” on page 23 in this manual.

Major Divisions of This Publication

This publication is divided into the following major parts.

Part One. Introduction to OGL/370

| This section is an introduction to the Overlay Generation Language/370 containing basic information about
| OGL/370 concepts and functions.

Part Two. Designing Overlays

This section is a tutorial introduction to the more commonly used features of OGL/370. Two sample overlays are presented. These sample overlays give you practical experience in writing commands and having them processed by the OGL/370 program.

- Chapter 2, “Designing a Simple Overlay” on page 23, contains information used to prepare overlays using boxes, pictures, words, and rules. A rule is a line that is either horizontal or vertical.
- Chapter 3, “Drawing Circles and Paths” on page 53, describes techniques used to create circles and paths on your overlay. A path consists of lines that go in whatever direction you choose. Paths can go horizontally, vertically, or diagonally. They can be used to construct arbitrary shapes on your overlay.
- Chapter 4, “Adding Text” on page 65, contains a more complete explanation of defining and placing text.
- Chapter 5, “Adding Graphics” on page 117, describes how to design, position, and place graphics.
- Chapter 6, “Adding Color” on page 131, describes how to add color to your document.

Part Two. Additional Features

This section describes more details about the OGL/370 commands presented in Part 2, “Designing Overlays” on page 21. It also describes some extra features that are used in more complex situations.

- Chapter 7, “Additional Features and Commands” on page 137, explains how figures, and groups of figures, can be placed and repeated in different places on your overlay. This chapter also explains how commonly used text can be stored in a symbolic dataset or symbolic file, and how to set default measurement units.
- Chapter 8, “Additional Features for Circles and Paths” on page 171, contains information about advanced features for use with paths and circles. It explains how to specify features of paths, such as shading and the degree of rounding of connections.

Part Three. Reference

This section is a reference guide for OGL/370. It contains three sections that provide a quick, convenient way of looking up the details of OGL/370.

- Chapter 9, “Overlay Generation Language Commands” on page 191, describes the types of commands, general rules for using the commands, and the syntax of the commands.
- Appendixes:
 - Appendix A, “Sample Overlay RCPT: Output Listing” on page 295, presents source listings for the “RCPT” overlay used as samples in this book
 - Appendix B, “Data-Set and File Allocation” on page 305, describes the attributes of MVS data sets, VSE files, and VM files that can be used with OGL/370.

- Appendix C, “System Dependent Information Procedures” on page 307, presents the job control language needed to access MVS data sets and VSE files for submitting overlays. It also shows the program invocation to be used in the VM environment.
- Appendix D, “Merging Overlays and Variable Data” on page 315, explains how to coordinate overlay size and position with the size and position specified in page definitions (PAGEDEF) and form definitions (FORMDEF). This appendix also correlates text specifications between OGL/370 and Print Management Facility (PMF).
- Appendix E, “The Symbolic Data Set and Symbolic File” on page 319, describes how to create a symbolic data set and a symbolic file.
- Appendix F, “Matching Fonts with Text Formatting” on page 323, describes how to select fonts to use with OGL/370.
- Appendix G, “Shade Patterns and Types” on page 325, provides samples of the two patterns and the 32 levels of shading that can be used with defined patterns, boxes, circles and paths.
- Appendix H, “Printer Characteristics” on page 335, explains the printing restrictions that apply to the printing subsystems supported by OGL/370.
- Appendix I, “OGL/370 Keywords” on page 337, lists the keywords of OGL/370 and describes how to modify them.
- Appendix J, “Storage Summary” on page 341, describes the storage requirements for printing an OGL/370 overlay with variable data.
- Appendix K, “Measurement Units Conversion Table” on page 343, provides tables for converting measurements in inches, millimeters, pels, and points.
- Appendix L, “Codes and Messages” on page 345, lists the text of messages that OGL/370 issues.
- The “Glossary” on page 395 contains definitions of terms used in this book.

Printers

Appendix H, “Printer Characteristics” on page 335 describes the different characteristics of IBM AFP printers and how they apply to OGL/370.

References to the 3800 Printing Subsystem refer to the 3800 Printing Subsystem Models 3, 6, and 8 (in compatibility mode) unless otherwise explicitly stated.

Note: OGL/370 does not support the 3800 Printing Subsystem Models 1 and 5.

References to the 3800 Printing Subsystem Model 3 and Model 8 also apply to the 3800 Printing Subsystem Model 6, unless otherwise explicitly stated.

References to generic logical device types apply to any IBM AFP printers compatible with them at the data stream level.

- I The 3800 Printing Subsystem Model and IBM AFP printers are used throughout this manual in examples. This does not imply that these are the only printers supported by OGL/370.

IBM page printers are mentioned throughout this manual. For more information about any of these devices, read the appropriate introduction and planning guide for the product that interests you.

Illustrations

Several of the illustrations in this book show OGL/370 overlays at larger than actual size to point out features of interest.

The shading patterns (and fonts) that OGL/370 generates may be represented differently on different printers. Shading patterns in illustrations are representative; you may not be able to reproduce them on your printer.

Related Publications

OGL/370: Getting Started, (G544-3691) is for the first-time OGL/370 user. It shows how to create boxes, circles, and patterns. It explains how to start OGL/370 and offers some coding hints.

OGL/370: Quick Reference, (S544-3703) is a guide for the experienced OGL/370 user. It summarizes the syntax of OGL/370.

OGL/370: Licensed Program Specifications, (G544-3697) provides the definitive statement of what OGL/370 does, and its hardware and software requirements.

OGL/370: Diagnosis Guide and Reference, (LH40-0208) is a reference book for system programmers.

ABOUT TYPE: IBM's Guide for Type Users, (G544-3122) describes the available fonts.

ABOUT TYPE: Technical Reference for Core Interchange Fonts, (S544-3708) describes the IBM Core fonts available with Print Services Facility (PSF) Version 2.

ABOUT TYPE: IBM's Technical Reference for 240-Pel Digitized Type, (S544-3516) assists you in preparing your overlay definition. This book provides information to help you use fonts with OGL/370.

Advanced Function Printing: Diagnosis Guide, (LH40-0201) describes how to calculate storage requirements.

Advanced Function Printing: Printer Information, (S544-3290) describes the characteristics of IBM's AFP printers.

IBM Page Printer Formatting Aid User's Guide, (S544-5284) describes form definitions and page definitions, and how to print an OGL/370 overlay with variable data from another program.

Advanced Function Printing Data Stream Reference, (S544-3202) describes the interchange data stream for Advanced Function Printing software.

Mixed Object Document Content Architecture Reference, (SC31-6802) describes the MO:DCA data stream.

Part 2. Designing Overlays

Chapter 2. Designing a Simple Overlay

When you finish this part, you will be able to write definitions for overlays using boxes, lines, pictures, and words. An example is shown below. However, such overlays can be designed more easily if you also read Chapter 7, “Additional Features and Commands” on page 137. More information about overlay design, including the specification of circles and paths, is contained in Chapter 8, “Additional Features for Circles and Paths” on page 171.

Note: Most of the examples used in this book have been developed for a medium overlay. The same examples could be used as page overlays by simply setting the offset values to 0,0.

The commands in this chapter are for the sample overlay shown in Figure 14, called “RCPT”.

DO NOT WRITE IN AREA BELOW

<div style="border: 1px solid black; width: 30px; height: 30px; margin: 0 auto;"></div> <p style="text-align: center; margin: 5px 0;">Total Units</p>	<p style="margin: 0;">F E E S</p>	<table border="1" style="width: 100%; height: 100%; border-collapse: collapse;"> <tr><td style="width: 50%; height: 20px;"></td><td style="width: 50%; height: 20px;"></td></tr> <tr><td style="width: 50%; height: 20px;"></td><td style="width: 50%; height: 20px;"></td></tr> <tr><td style="width: 50%; height: 20px;"></td><td style="width: 50%; height: 20px;"></td></tr> <tr><td style="width: 50%; height: 20px;"></td><td style="width: 50%; height: 20px;"></td></tr> </table>									<p>Registration</p> <p>Out-of-State</p> <p>Other</p> <p>Total</p>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; border-bottom: 1px solid black;">LAST NAME</td> <td style="width: 33%; border-bottom: 1px solid black;">FIRST NAME</td> <td style="width: 33%; border-bottom: 1px solid black;">MI</td> </tr> <tr> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td></td> </tr> <tr> <td colspan="3" style="border-bottom: 1px solid black; text-align: center;">SOCIAL SECURITY NUMBER</td> </tr> <tr> <td colspan="3" style="border-bottom: 1px solid black; height: 20px;"></td> </tr> <tr> <td style="border-bottom: 1px solid black;">Received by</td> <td colspan="2" style="border-bottom: 1px solid black;">Date</td> </tr> <tr> <td colspan="3" style="text-align: center; border-bottom: 1px solid black;"><i>Robinson K. Russo</i></td> </tr> <tr> <td colspan="3">Robinson K. Russo</td> </tr> <tr> <td colspan="3">President</td> </tr> </table>	LAST NAME	FIRST NAME	MI	-	-		SOCIAL SECURITY NUMBER						Received by	Date		<i>Robinson K. Russo</i>			Robinson K. Russo			President			<p style="margin: 0;">Keep this copy. NOT VALID as ID.</p> <p style="font-size: 2em; margin: 0;">N O T I C E</p>
LAST NAME	FIRST NAME	MI																																			
-	-																																				
SOCIAL SECURITY NUMBER																																					
Received by	Date																																				
<i>Robinson K. Russo</i>																																					
Robinson K. Russo																																					
President																																					

TropiCal
Community College

Figure 14. Overlay RCPT

This is the bottom portion of the larger overlay shown in Figure 1 on page 3. In this chapter, we treat the bottom portion as a separate overlay. A helpful way to learn how to use OGL/370 is to enter the commands yourself. By doing so, you will write an overlay definition that can produce this form.

Conventions

As each command is presented, you are shown an example similar to Figure 15. The user entry appears on the top line; the description of each entry appears in lowercase letters under the entry.

OVERLAY	rcpt	SIZE	7.25 in	3.25 in	OFFSET	0	.5 in	;
command	name	subcmd	width	height	subcmd	coordinates		end
word		word			word	horizontal	vertical	marker

Figure 15. Command Sample

The examples have four types of entries:

KEYWORDS Command and subcommand keywords must be spelled exactly as they are shown, you may use uppercase or lowercase. For example, in the **OVERLAY** command, you must spell the subcommand keyword, **SIZE**, just as shown; SIZES is not acceptable.

Note: In this manual, command and subcommand keywords are shown in **BOLD UPPERCASE** letters.

DEFAULTS Parts of some commands, for example the thickness of a rule, have defaults and therefore do not have to be entered. If nothing is typed, OGL/370 automatically chooses (defaults to) one for you. For example, if rule thickness is not specified, OGL/370 defaults to **MEDIUM** thickness. See Figure 16 on page 26 for an example of how defaults are displayed in the command examples contained in this manual.

Note: In this manual, defaults are shown in **UNDERLINED BOLD UPPERCASE**.

values You can choose some words or numbers for the particular overlay you are designing. In the example above, we chose "RCPT" for the overlay name and "7.25 in" by "3.25 in" for the overlay size.

Note: In this manual, values are shown in lowercase.

Each heading that introduces a new command or a new use of a command includes the name of the command in parentheses.

Writing Comments

Comments are optional. They can be used in an overlay definition as reminders or as explanations about definition parts. Comments may appear before or after a command (or group of commands) or even inside a command. They are not part of the definition itself and do not affect or appear on the overlay.

Note: A single comment can extend over more than one line. If a comment appears on a line after the end-marker semicolon (;), you should not continue the comment on the next line. You can, of course, begin a new comment on the next line.

There are two types of comments:

One-word comments

The word must immediately follow a hyphen (-). Do not put a space between the hyphen and the word. In the following example, “-SIGNATURE” is used as a comment to identify the signature line of the overlay definition:

```
DRAWRULE -SIGNATURE 2.5 IN;
```

Multiple-word comments

The words must be enclosed by apostrophes (') and immediately follow a hyphen. Do not put a space between the hyphen and the first apostrophe. Because there are two signature lines on the sample overlay, you might want to comment the “president’s signature” line like this:

```
DRAWRULE -'PRESIDENT''S SIGNATURE' 2.5 IN;
```

Note: An apostrophe within a comment is represented by two apostrophes ('); a semicolon (;) within a comment is represented by two semicolons (;:).

When you write a comment, you must begin and end the entire comment with one or more blank spaces. You can begin any line with a hyphen, as long as the previous line ends with a blank.

How the comments are used, is up to you. Here are some suggestions:

- Comments should make the overlay definition clearer to the reader. Adding comments to simple commands might actually make the definition harder to read. For example, the following command should be clear to anyone who is familiar with the language and probably does not need to be commented:

```
OVERLAY RCPT SIZE 7.25 IN 3.25 IN OFFSET 0 IN .5 IN;
```

- Comments should not make important data hard to find.
- Comments are useful as headings to show the organization of the definition.

Getting Started

This section introduces four commands: **CONTROL**, **OVERLAY**, **ORIENT**, and **DRAWMASK**. At the end of this section, you should be able to print a sample overlay and a definition listing.

Specifying Storage, Message, and SOSI Options (CONTROL)

Overlays can be stored in libraries and are then available for printing when you tell the computer to retrieve them. You do this with the **CONTROL** command, which should be the first command in an overlay definition. In the same command, you can specify which message types to include in your listing, how double-byte text should be handled, and whether you want a summary of information to help you determine the overlay storage requirements

Note: Only one **CONTROL** command is allowed in each overlay definition.

The **CONTROL** command has the following parts:

CONTROL	NOSTORE	ALL	NOSUMMARY	SOSI	;
command	storage	messages	summary	sosi	end
word				option	marker

Figure 16. **CONTROL** Command

command word

CONTROL

storage

Choose from:

NOSTORE

The overlay is not stored (default).

STORE

The overlay is stored (providing that one with the same name does not already exist).³ In MVS and VSE, the overlay is stored in the specified library.²In VM, the overlay is stored on your A-disk. Choose this option when the definition is completed and ready for use.

REPLACE

The overlay replaces an identically named overlay that is already stored. The overlay that is replaced is no longer available for use. This option is usually used when you have completed changes or revisions to an existing overlay.

messages

An explanation of the message types is given in "Messages" on page 6. Choose from:

ALL

Informational, warning, and error messages are printed (default).

WARN

Warning and error messages are printed.

ERROR

Only error messages are printed.

summary

If you ask for it, OGL/370 compiles a summary report of statistics about your overlay. These statistics can be used to determine how much printer storage the overlay needs. This information is not important while you are learning to use the language; therefore, we have included the explanation of this report in Appendix J, "Storage Summary" on page 341. Choose from:

1. When used in a printing job, the overlay is copied from library storage into printer storage. The summary contains information that relates only to printer storage and is of particular value if you are printing your overlay on an IBM 3800 Printing Subsystem Model 3 or Model 8.

2. See Appendix C, "System Dependent Information Procedures" on page 307.

3. If the **STORE** or **REPLACE** option is specified in the **CONTROL** command, OGL/370 adds the prefix "O1" to the overlay name specified in the **OVERLAY** command and stores the overlay in a library under the full name (prefix + specified name).

NOSUMMARY

The summary report is not included (default).

SUMMARY

The summary report is included.

sosi option Choose from:

SOSI All DBCS text must be surrounded by SOSI delimiters.

NOSOSI All DBCS text **must not** be surrounded by SOSI delimiters.

Note: For more information on using the *sosi* option, see Chapter 4, “Adding Text” on page 65.

end marker Always end a command with an end marker (;).

Figure 16 on page 26 illustrates the **CONTROL** command for overlay “RCPT”; which is not stored, requests all messages, requests no summary, and indicates that double-byte text must be surrounded by SOSI delimiters.

Beginning the Overlay (OVERLAY)

The **OVERLAY** command names the overlay and specifies its size and placement.

Before you design an overlay, you need to decide whether you will be using it as a medium overlay or a page overlay. For more information on medium overlays and page overlays, see “Medium and Page Overlays” on page 6. When you are designing an overlay and you specify its size in the **OVERLAY** command, it is helpful to be generous. For example, if text in a particular font is too big to fit on the overlay, none of the text is printed. However, if you define the overlay larger than the final version, the text is printed. Accordingly, we have defined overlay “RCPT” to be ½-inch wider and ¼-inch higher than it appears on the paper.

Similarly, if you want a frame around an overlay like the one in Figure 14 on page 23, the overlay size that you specify in this command must be larger than the frame itself. This is further explained in “Defining the Box (**DRAWBOX**)” on page 41.

After the printed overlay appears the way you want, be sure to change the **OVERLAY** command so it specifies the desired size of the overlay. As we mentioned earlier, variable data can be printed with the overlay; a set of instructions, called a page definition, places that data. Among other things, the page definition specifies the size of the page. It is important that the page size specified in the page definition match the overlay size specified in the **OVERLAY** command. If these sizes do not match, the variable data might not be placed correctly on the overlay.

The **OVERLAY** command has the following parts:

OVERLAY	rcpt	SIZE	7.25 in	3.25 in	OFFSET	0	.5 in	;
command	name	subcmd	overlay	overlay	subcmd	coordinates		end
word		word	width	height	word	horizontal	vertical	marker

Figure 17. **OVERLAY** Command

command word

OVERLAY

name For the sample overlay, we have chosen the name “RCPT”, for “receipt”.

Overlay names must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

subcommand word

SIZE

overlay width and height

The printer sees all overlays as rectangles. Therefore, you must specify width and height. Use a number (*n*) and one of the following units of measurement:

n

IN	Inches
MM	Millimeters
PELS	Pels

Notes:

1. Enter two measurements; the first one is for width and the second one is for height.
2. Different units of measurement can be used in the same command. For example, the width of the overlay can be specified as 7.25 inches and its height as 780 pels.

subcommand word

OFFSET

This subcommand tells where to place the overlay. You specify offset values as horizontal and vertical coordinates.

horizontal and vertical coordinates

You define the offset using coordinates. For a medium overlay, the coordinates are added to the medium origin. For a page overlay, the coordinates are added to the IPO coordinates to position the overlay on the logical page.

n

IN	Inches
MM	Millimeters
PELS	Pels

Notes:

1. Enter two coordinates; the first one is for horizontal and the second one is for vertical.
2. See Figure 20 on page 31 to better understand the origins.

end marker Always end a command with an end marker (;).

Figure 17 on page 27 shows the size and origin specifications for overlay "RCPT" as it appears for printing on the IBM 3800 Model 3. Notice that the horizontal distance is given as 0. Because 0 inches is the same as 0 millimeters or 0 pels, you do not need to specify **IN**, **MM**, or **PELS**.

Orienting the Overlay (ORIENT)

The **ORIENT** command specifies the orientation of an overlay relative to the paper.

To specify the orientation of an overlay, you must understand what is meant by the "top of the paper" and the "top of the overlay". The meaning of orientation depends on the type of printer you use to print the overlay. Refer to your printer publications for information about "top of the overlay".

The "top of the overlay" refers to the top of the overlay in the reading position. See Figure 18 on page 29.

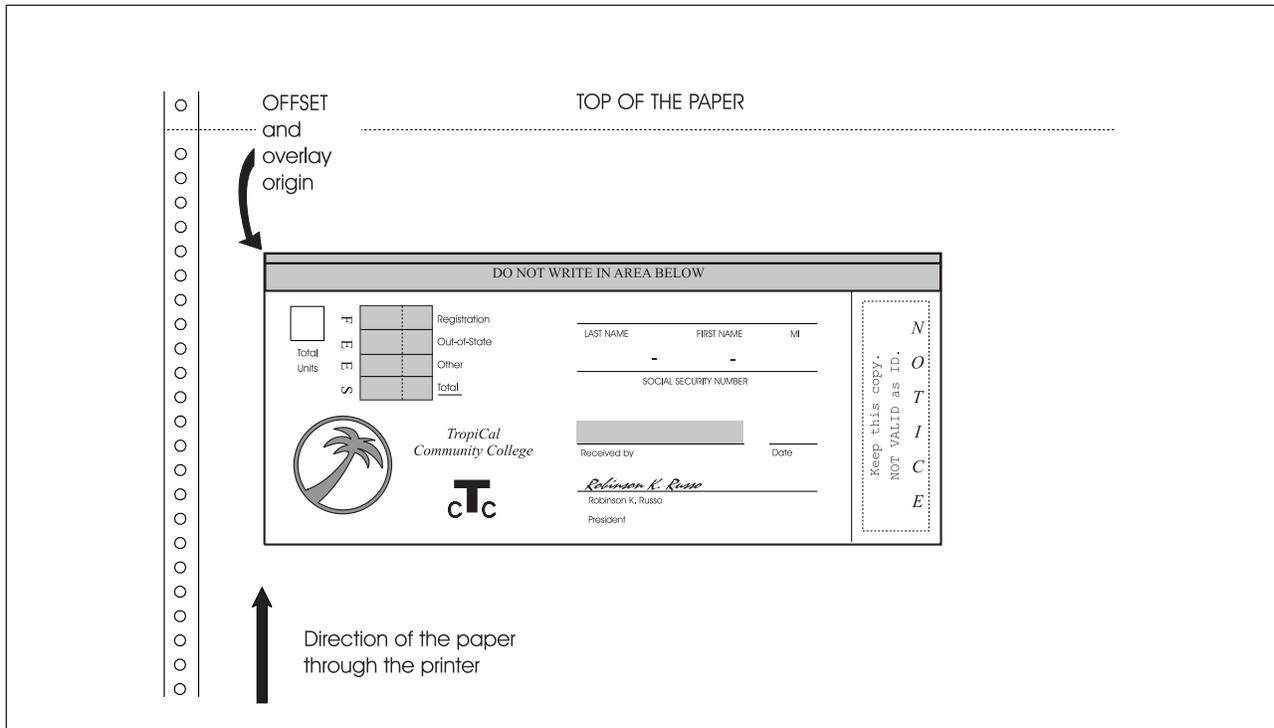


Figure 18. Overlay Specified with 0° Orientation for the IBM 3800. The artwork used for the palm tree was created as a separate page segment at the 0° orientation.

In Figure 18, the top of the overlay and the top of the paper are on the same side. The orientation of the overlay is 0°. Most overlays are printed in this orientation. In some cases, however, it might save paper to print an overlay in another orientation. See Figure 19 on page 30 and Figure 20 on page 31 for illustrations of the 90° orientation.

Note: Your printer may not be able to print text in all four orientations. Some restrictions are explained in Appendix H, "Printer Characteristics" on page 335. Refer to your printer publications for detailed information.

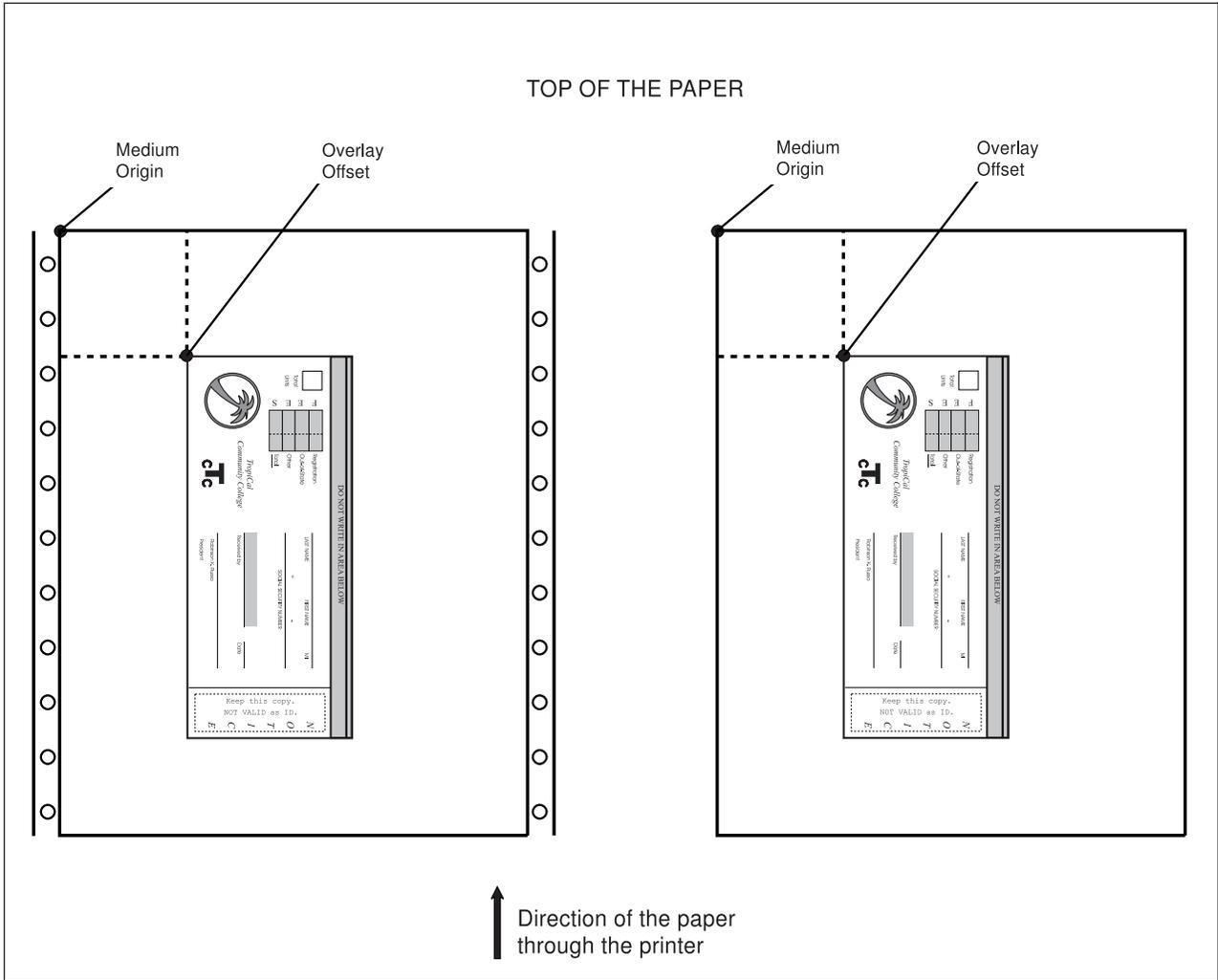


Figure 19. Medium Overlay Specified with 90° Orientation. The artwork used for the palm tree was created as a separate page segment at the 90° orientation.

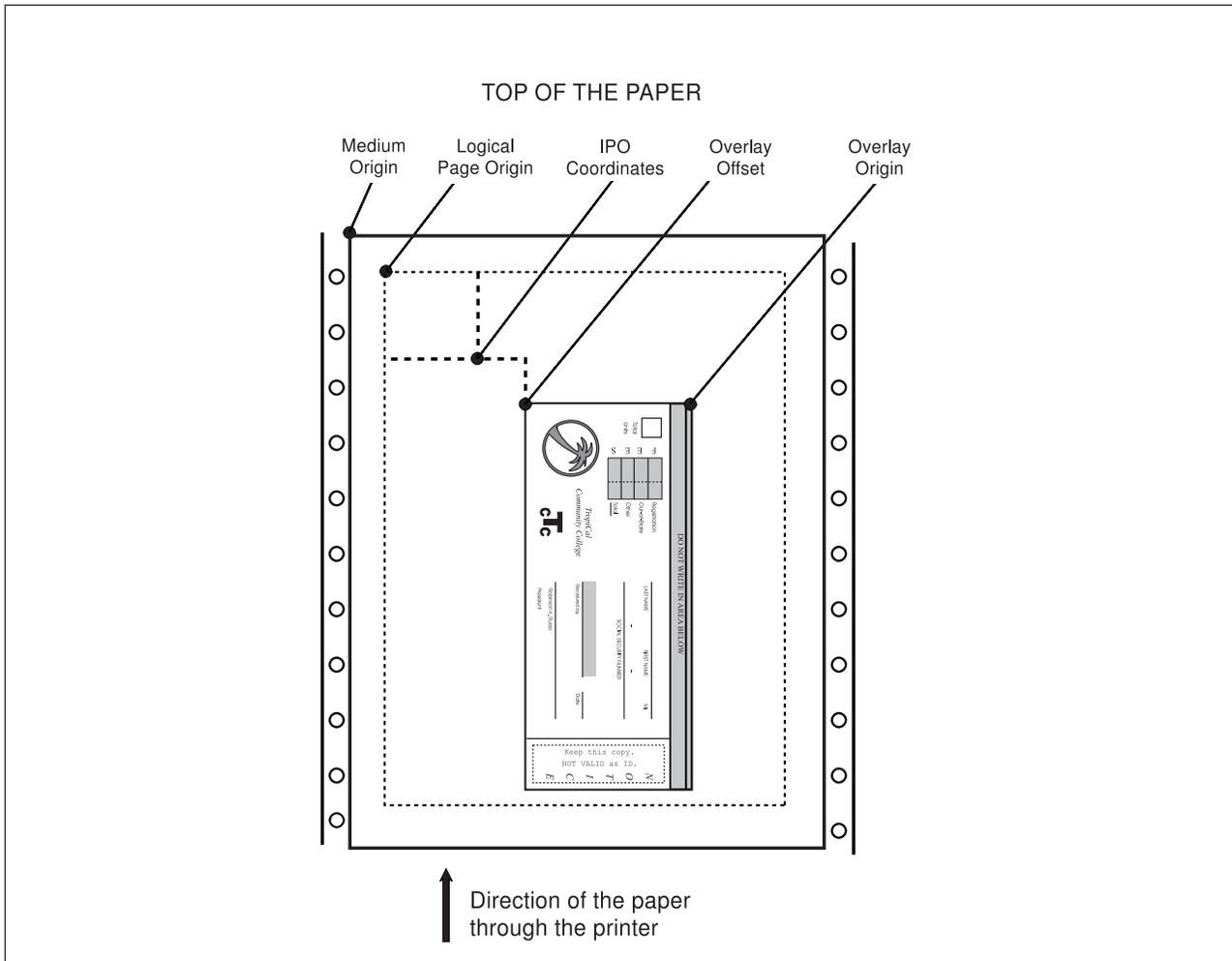


Figure 20. Page Overlay Specified with 90° Orientation. Shown on a 3800 Printer. The artwork used for the palm tree was created as a separate page segment at the 90° orientation.

Figure 21 shows that there are only two entries for the **ORIENT** command:

```
ORIENT    0    ;
command  orientation  end
word      marker
```

Figure 21. **ORIENT** Command

command word

ORIENT

orientation Choose from:

- 0** (Default)
- 90**
- 180**
- 270**

end marker Always end a command with an end marker (;).

Figure 21 shows the **ORIENT** command for the sample overlay "RCPT". The **ORIENT** and the **OVERLAY** commands are closely related. If you change the orientation of an overlay, you may also want to change

OFFSET in the **OVERLAY** command. Otherwise, the result might not be what you expect, depending on the paper size and the overlay size.

```
OVERLAY XMPLE SIZE 10 IN 6 IN OFFSET 2 IN 4 IN; ORIENT 0;
```

If you change the orientation to 90°, the result places the overlay as represented by the dotted rules. This is off the *printable area* of the paper.

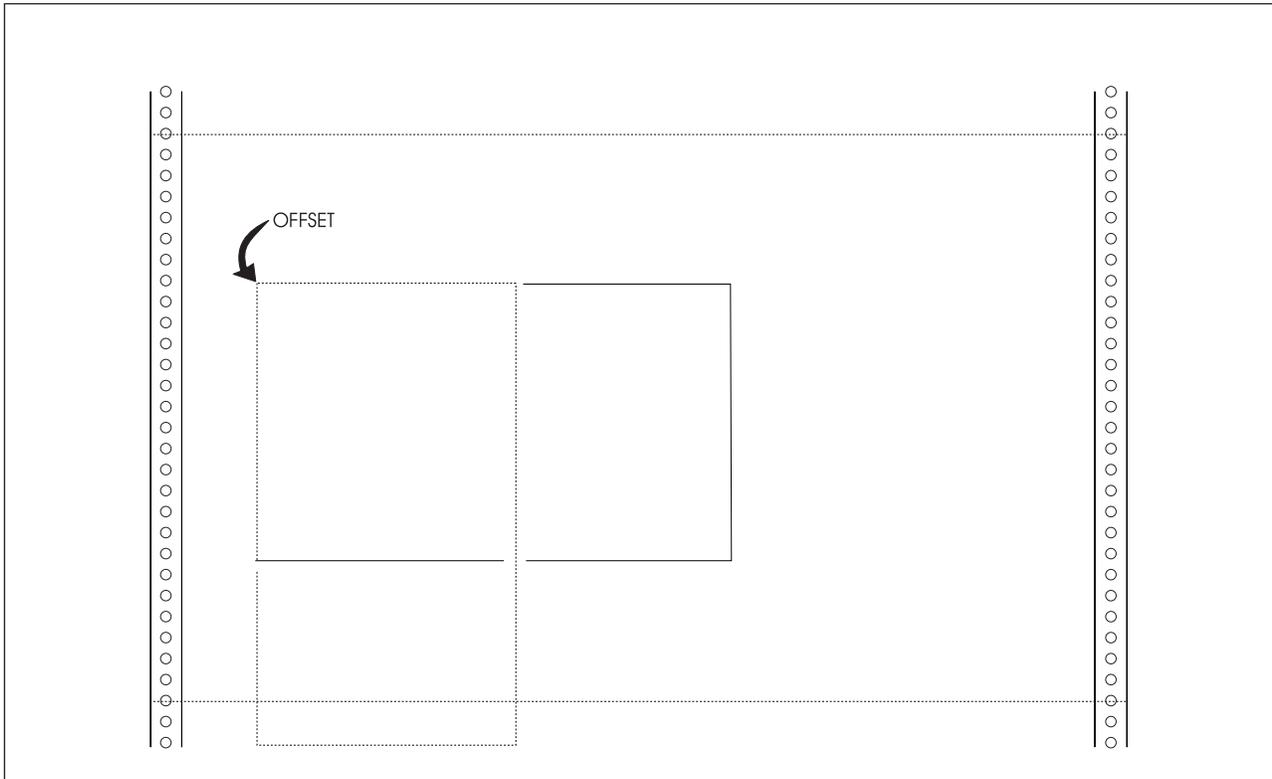


Figure 22. **OFFSET** Subcommand and Orientation. Shown on paper used in the IBM 3800.

This command enables the overlay “XMPLE” to be rotated 90° and remain in the printable area:

```
OVERLAY XMPLE SIZE 10 IN 6 IN OFFSET 2 IN 1 IN;  
ORIENT 90;
```

Drawing a Grid (DRAWMASK)

It is easier to design an overlay on a *mask*, a grid of equally spaced vertical and horizontal rules. A mask helps you place rules, boxes, text, and graphics. The **DRAWMASK** command allows you to include such a mask on your overlay.

Here are a few observations about rule spacing:

- Rules too close together or too far apart are not helpful.
- You can choose different vertical and horizontal spacings.
- Generally, the rule spacing should be in the same unit of measurement used to design the overlay.

Figure 23 shows the **DRAWMASK** command with the following parts:

DRAWMASK	.25 in	.25 in	;
command word	first spacing	second spacing	end marker

Figure 23. **DRAWMASK** Command

command word

DRAWMASK

first spacing

This is the horizontal distance from the left side of one vertical rule to the left side of the next. See “PART A” of the illustration in Figure 24 on page 34.

Specify a number (*n*) and one of the following units of measurements:

n

IN Inches
MM Millimeters
PELS Pels

second spacing

This is the vertical distance from the top of one horizontal rule to the top of the next. See “PART B” of the illustration in Figure 24 on page 34.

Specify a number (*n*) and one of the following units of measurement:

n

IN Inches
MM Millimeters
PELS Pels

Note: The rules are drawn relative to the overlay, rather than to the paper. For example, the vertical rules of an overlay oriented at 90° are drawn across the paper, but down the overlay.

end marker

Always end a command with an end marker (;).

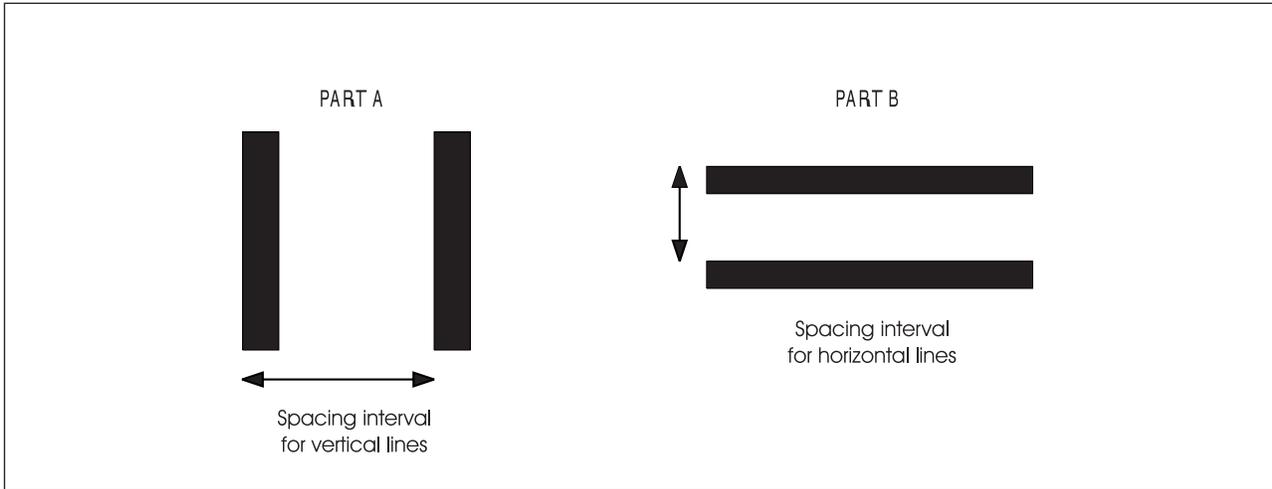


Figure 24. Spacing Intervals for a Grid. (Not actual size.)

For overlay “RCPT”, we want a mask of rules at $\frac{1}{4}$ -inch intervals, both vertically and horizontally. Write the command and compare your version with Figure 23 on page 33.

So far, this is what the “RCPT” overlay definition looks like:

```

-'GETTING STARTED'
CONTROL NOSTORE ALL NOSUMMARY;
OVERLAY RCPT SIZE 7.25 IN 3.25 IN OFFSET 0 .5 IN;
ORIENT 0;
DRAWMASK .25 IN .25 IN;

```

Note: The **OVERLAY** command and, unless you omit them, the **CONTROL** and **ORIENT** commands should be the first three commands in an overlay definition. “Using Defaults and Abbreviations” on page 51 explains when **CONTROL** and **ORIENT** can be omitted.

After processing this OGL/370 definition, your printout should look like the illustration in Figure 25. Remember, the mask is $\frac{1}{4}$ -inch wider and $\frac{1}{4}$ -inch higher than the overlay shown in Figure 14 on page 23 because you have defined your overlay $\frac{1}{4}$ -inch wider and $\frac{1}{4}$ -inch higher than the frame of the “RCPT” sample.

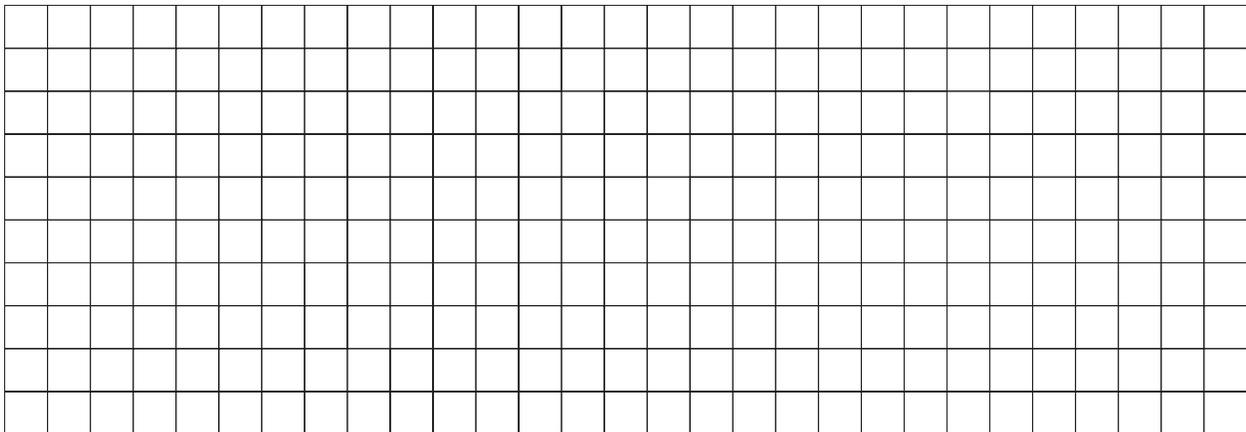


Figure 25. The Mask for Overlay RCPT

Drawing Rules

You now have a mask that makes placing rules, boxes, text, and graphics easier. We begin by placing the rules as follows:

- Use the **POSITION** command to specify where the rule begins.
- Use the **DRAWRULE** command to describe the rule.

Here are two general principles of OGL/370:

- To position almost anything on an overlay, you must use a **POSITION** command.
- A **POSITION** command must precede what it is positioning.

Use the **DRAWRULE** command to draw rules that are not part of a box; you use another command, **DRAWBOX**, to draw boxes.

Figure 26 shows the rules that are drawn with **DRAWRULE**. The rule numbers in the figure are explained next.

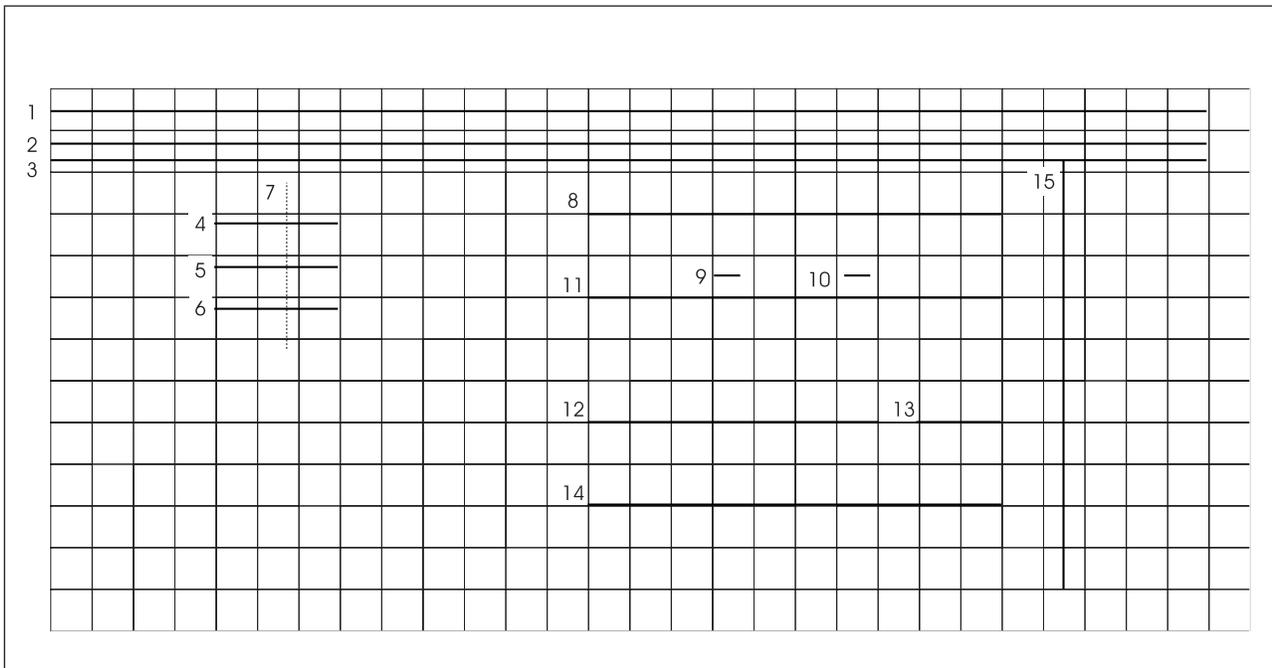


Figure 26. Rules

Where to begin positioning rules and how to continue is partly a matter of personal choice. You might want to change the overlay. For example, you may want to add or delete some rules. The numbers in Figure 26 show the order in which we defined the rules.

Here are some guidelines for drawing rules and boxes and for placing text and graphics:

- Define all rules in one set, all boxes in another, and so on. This method makes it easier to find the definition of a particular item.
- Within each set, define the items in an understandable order.

These are guidelines rather than rules; strictly following an order is not always possible or even desirable. In Figure 26, we depart a bit from the procedure (for example, with rules 4, 5, and 6 since they seem to go together). Also, when you know all the commands, you might want to define a collection of rules, boxes, and text as one unit.

You are now ready to begin positioning and defining the rules.

Positioning the Rule (POSITION)

The **POSITION** command specifies the coordinates of a point relative to the overlay, (that is, the horizontal and vertical distances from the overlay origin). Look at Figure 26 on page 36. What are the coordinates of the beginning of rule 1? Of rule 4?

Note: The beginning of a rule drawn across an overlay is the left end. The beginning of a rule drawn down an overlay is the top end.

Rule 1 begins at the left margin of the overlay and 0.1 inches from the top margin. Therefore, its horizontal distance is 0 and its vertical distance is 0.1 inches. The coordinates of rule 1 are 0 inches and 0.1 inches.

Rule 4 begins 1 inch from the left margin and 0.8 inches from the top margin. Its coordinates are 1 inch and 0.8 inches.

Figure 27 shows the **POSITION** command for rule 4. The following parts are specified in the **POSITION** command:

POSITION	<u>ABSOLUTE</u>	1 in	<u>ABSOLUTE</u>	.8 in	;
command word	origin option	first coordinate	origin option	second coordinate	end marker

Figure 27. **POSITION** Command

command word

POSITION

origin option This is the point from which you measure.⁵ Measurement from the overlay origin is indicated by this option. Choose from:

ABSOLUTE

LEFT

RIGHT

UP

DOWN

first coordinate When the origin option is ABSOLUTE, the first coordinate is always the horizontal coordinate. Specify a number (*n*) and a unit of measurement. Choose from:

n

IN Inches

MM Millimeters

PELS Pels

origin option This is the point from which you measure. When you measure from the overlay origin, it is indicated by this option. Choose from:

ABSOLUTE

LEFT

RIGHT

UP

DOWN

4. You do not always have to measure from the overlay origin. However, it is usually easiest to do so. Later in this chapter you will learn other ways to specify the coordinates.

5. You do not always have to measure from the overlay origin. However, it is usually easiest to do so. Later in this chapter you will learn other ways to specify the coordinates.

second coordinate

When the origin option is **ABSOLUTE**, the second coordinate is always the vertical coordinate. Specify a number (*n*) and a unit of measurement. Choose from:

- n*
- IN** Inches
- MM** Millimeters
- PELS** Pels

end marker Always end a command with an end marker (;).

Defining the Rule (DRAWRULE)

To describe a rule, tell the printer the direction, length, thickness, and type of rule as shown in Figure 28.

DRAWRULE	<u>ACROSS</u>	.75 in	LIGHT	<u>SOLID</u>	;
command word	direction	length	thickness	rule type	end marker

Figure 28. **DRAWRULE** Command

command word

DRAWRULE

rule direction Choose from:

- ACROSS A horizontal rule, relative to the overlay (default).
- DOWN A vertical rule, relative to the overlay.

rule length Enter a number (*n*) and a unit of measurement. Choose from:

- n*
- IN** Inches
- MM** Millimeters
- PELS** Pels

rule thickness Either choose a ready-made rule or customize a rule by specifying a value for its thickness in pels. Choose from:

- LIGHT
- MEDIUM (Default)
- BOLD
- n* Thickness in pels, do not add the word “pels”.

Figure 29 illustrates ready-made rules. To help you custom-build rules, the pel thickness of each rule is given in parentheses.

LIGHT (2 pels)	_____
MEDIUM (4 pels)	_____
BOLD (6 pels)	_____

Figure 29. Rule Thicknesses

rule type Choose from:
SOLID (Default)

DASHED
DOTTED

Figure 30 illustrates rule types for a 4-pel (**MEDIUM**) rule.

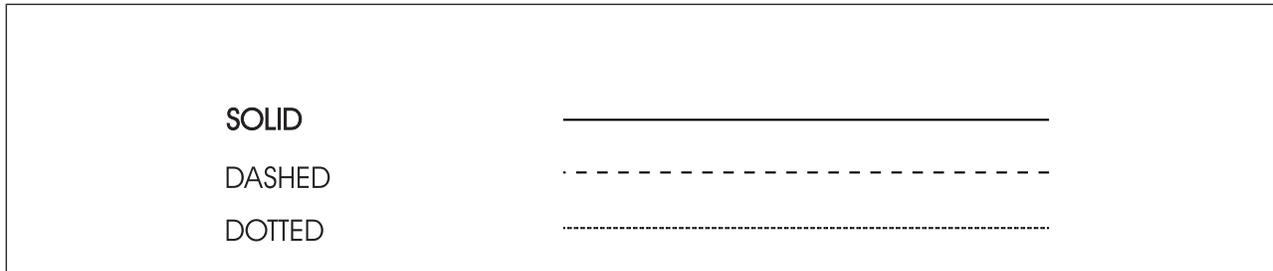


Figure 30. Rule Types for a 4-Pel Rule

Note: Problems can occur when printing an overlay that contains many dashed or dotted rules. See “Printer Storage Limitations” on page 335 and Appendix J, “Storage Summary” on page 341 for further details.

You are now ready to write the **DRAWRULE** command. Figure 28 on page 38 contains the command defining rule 4, from Figure 26 on page 36.

Write the commands to position and describe the numbered rules from Figure 26 on page 36. Use inches for all measurements. The **POSITION** command must immediately precede the **DRAWRULE** command for each rule it positions. When you are done, compare your results with Figure 31 on page 40.

```

-'DRAWING RULES'
  POSITION ABSOLUTE 0 ABSOLUTE .1 IN;
  DRAWRULE ACROSS 7 IN MEDIUM SOLID;
  POSITION ABSOLUTE 0 ABSOLUTE .3 IN;
  DRAWRULE ACROSS 7 IN MEDIUM SOLID;
  POSITION ABSOLUTE 0 ABSOLUTE .4 IN;
  DRAWRULE ACROSS 7 IN MEDIUM SOLID;

  POSITION ABSOLUTE 1 IN ABSOLUTE .8 IN;
  DRAWRULE ACROSS .75 IN LIGHT SOLID;
  POSITION ABSOLUTE 1 IN ABSOLUTE 1.05 IN;
  DRAWRULE ACROSS .75 IN LIGHT SOLID;
  POSITION ABSOLUTE 1 IN ABSOLUTE 1.3 IN;
  DRAWRULE ACROSS .75 IN LIGHT SOLID;
  POSITION ABSOLUTE 1.45 IN ABSOLUTE .55 IN;
  DRAWRULE DOWN 1 IN LIGHT DASHED;

  POSITION ABSOLUTE 3.25 IN ABSOLUTE .75 IN;
  DRAWRULE ACROSS 2.5 IN MEDIUM SOLID;

  POSITION ABSOLUTE 3.25 IN ABSOLUTE 1.25 IN;
  DRAWRULE ACROSS 2.5 IN MEDIUM SOLID;
  POSITION ABSOLUTE 4 IN ABSOLUTE 1.1 IN;
  DRAWRULE ACROSS .15 IN MEDIUM SOLID;
  POSITION ABSOLUTE 4.8 IN ABSOLUTE 1.1 IN;
  DRAWRULE ACROSS .15 IN MEDIUM SOLID;

  POSITION ABSOLUTE 6.1 IN ABSOLUTE .4 IN;
  DRAWRULE DOWN 2.6 IN MEDIUM SOLID;

  POSITION ABSOLUTE 3.25 IN ABSOLUTE 2 IN;
  DRAWRULE ACROSS 1.75 IN MEDIUM SOLID;

  POSITION ABSOLUTE 5.25 IN ABSOLUTE 2 IN;
  DRAWRULE ACROSS .5 IN MEDIUM SOLID;

  POSITION ABSOLUTE 3.25 IN ABSOLUTE 2.5 IN;
  DRAWRULE ACROSS 2.5 IN MEDIUM SOLID;

-'HEADING RULES'

-'FEES RULES'

-'NAME RULE'

-'SOC. SEC. RULES'

-'NOTICE RULE'

-'RECEIVED RULE'

-'DATE RULE'

-'PRESIDENT RULE'

```

Figure 31. **DRAWRULE** Commands for Overlay RCPT

You are now ready to print this version of the overlay. If you have not already done so, add the **POSITION** and **DRAWRULE** commands that you have just written to your overlay definition for “RCPT”. After OGL/370 processes the definition, your printout should look like the illustration in Figure 26 on page 36, without the numbers.

Drawing Boxes

Many overlays include boxes. In fact, many overlays are completely enclosed by a box, as is overlay RCPT. It takes four **POSITION/DRAWRULE** command pairs to draw a box. It takes only one **POSITION/DRAWRULE** pair to draw a box. Therefore, you should use **DRAWBOX** whenever possible. Figure 32 illustrates boxes in overlay “RCPT”.

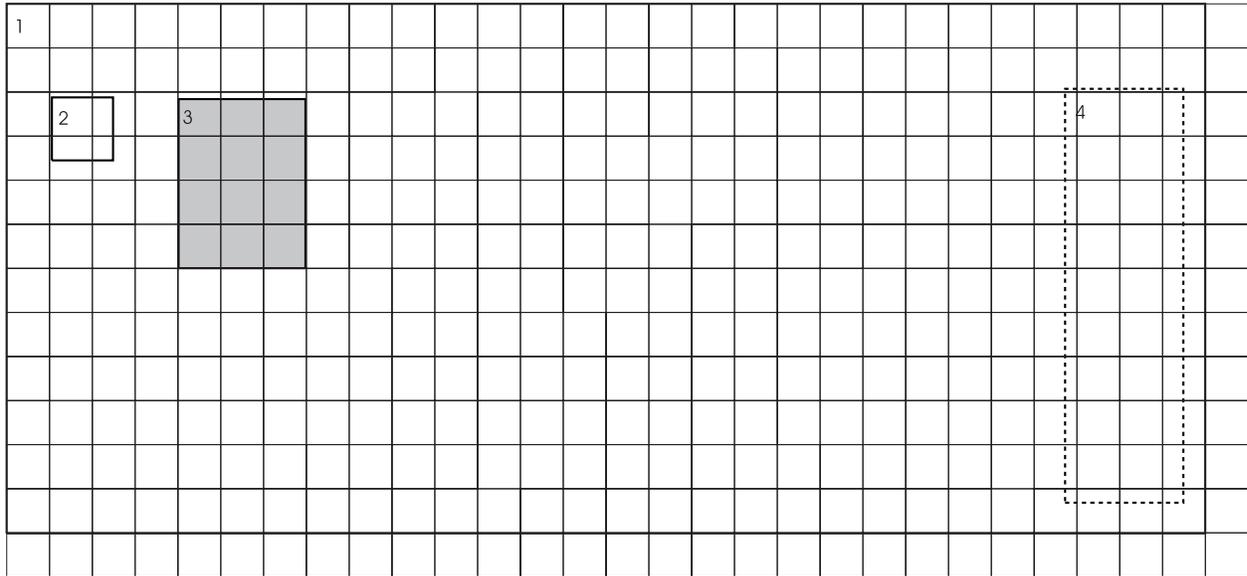


Figure 32. Boxes

There are two steps to draw a box:

1. Use a **POSITION** command to specify where the box begins.
2. Use a **DRAWBOX** command to describe the box.

Positioning the Box (POSITION)

In Figure 32, each box number is at the box origin.

You have already used the **POSITION** command for rules. Use the **POSITION** command to position Box 3 in Figure 32. Compare your version with Figure 33. Refer to page Figure 27 on page 37 if needed.

POSITION	ABSOLUTE	1 in	ABSOLUTE	.55 in	;
command	origin	first	origin	second	end
word	option	coordinate	option	coordinate	marker

Figure 33. **POSITION** Command

Defining the Box (DRAWBOX)

Look at Box 3 in Figure 32. To describe the box to someone who had not seen it, what would you say? You would probably mention the size (width and height), the border (from **DRAWRULE** you know that rules have a thickness and a type), and the shading. Figure 34 on page 42 illustrates how to specify that information for the printer.

DRAWBOX	.75 in	1 in	MEDIUM	SOLID	SHADE	STANDARD	LIGHT	;
command word	width	height	border thickness	border type	shading option	shade pattern	shade type	end marker

Figure 34. **DRAWBOX** Command

Note: You do not need to specify **MEDIUM**, **SOLID**, or **STANDARD** because they are the defaults. The other examples in this book have specified the defaults as a reminder that the defaults are in effect.

command word

DRAWBOX

Size

Box width is measured from the left side of one rule to the left side of the next. Box height is measured from the top of one rule to the top of the next. For example, if you specify a box that is $\frac{1}{2}$ -inch wide and $\frac{1}{4}$ -inch high with a **MEDIUM** border thickness (4 pels), you get a box with the dimensions shown in Figure 35.

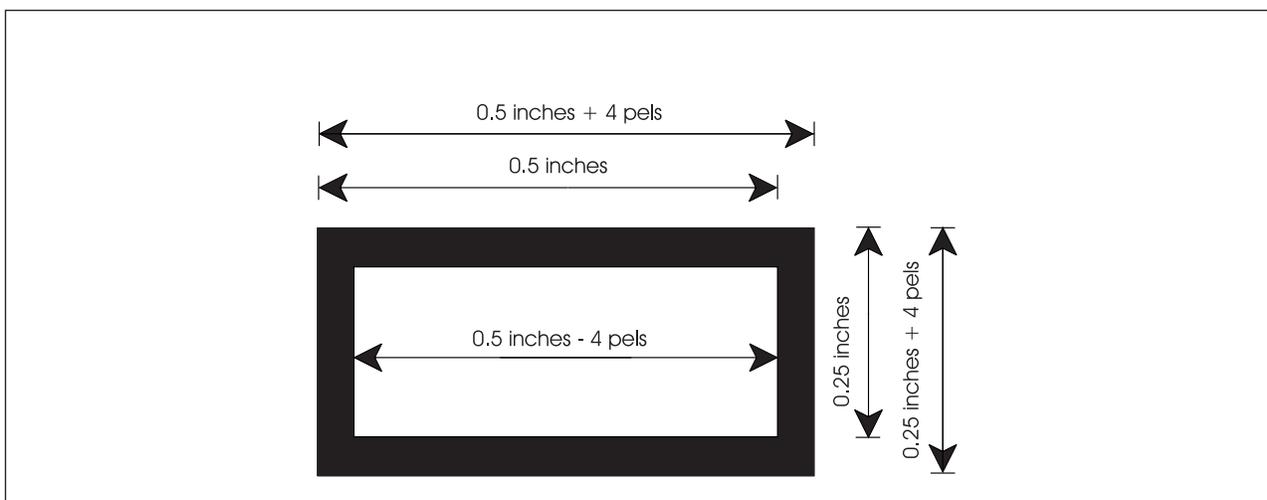


Figure 35. Box Size. (Not actual size.)

The result is a box that is the size specified plus the thickness of one border. For most overlays (including this one), the additional size is not critical (4 pels equal only $\frac{1}{60}$ of an inch). However, if you wish to place a box so it is almost, but not quite, touching another rule or box, the exact size is important. See Figure 36 on page 43 for more information.

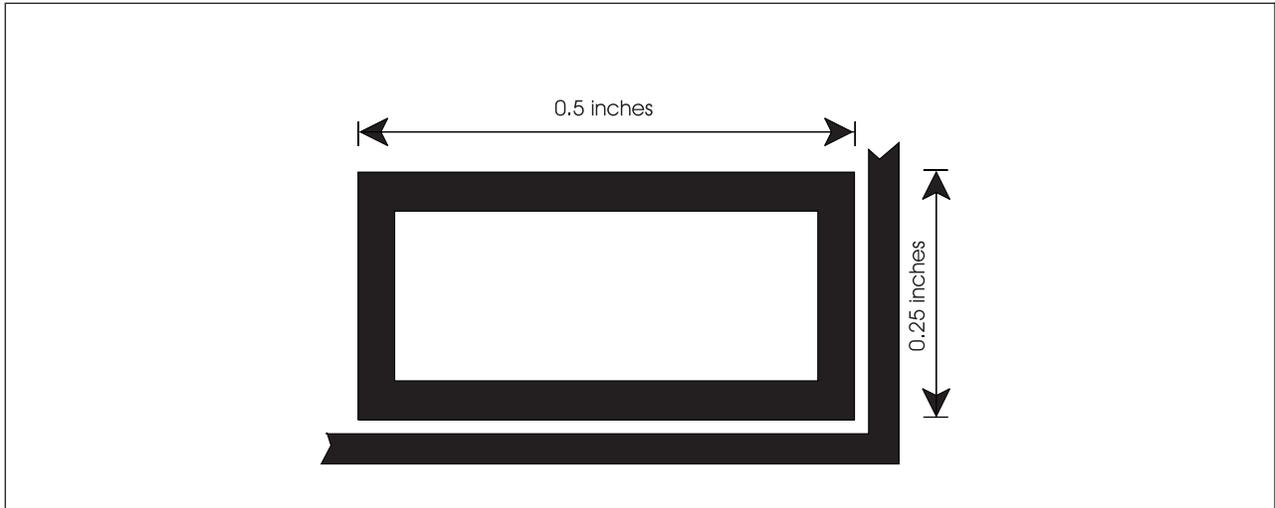


Figure 36. Exact Box Placement. (Not actual size.)

1. Convert the desired box size to pels: 0.25 inches equals 60 pels.
2. Subtract the border thickness, which in this example is 4 pels (see the next entry for the standard border thicknesses).

Thus, to draw a box that is exactly 0.5 inches by 0.25 inches measured on the outside of the border, you specify a box 116 pels by 56 pels.

width and height

To specify the box width and box height, enter a number (*n*) and a unit of measurement. Choose from:

<i>n</i>	IN	Inches
	MM	Millimeters
	PELS	Pels

Note: Enter two measurements; one for *width* and one for *height*.

Border:

border thickness

Describing border thickness is the same as describing rule thickness in **DRAWRULE**. See Figure 29 on page 38 for thicknesses. Either choose a ready-made border or customize a border by specifying a value for its thickness in pels. Choose from:

LIGHT	2 pels thick.
MEDIUM	4 pels thick (default).
BOLD	6 pels thick.
<i>n</i>	Thickness in pels, do not add the word "pels".

border type

The border types are also the same as those in the **DRAWRULE** command. See Figure 30 on page 39 for types. Choose from:

- SOLID** (Default)
- DASHED**
- DOTTED**

Note: There may be problems in printing an overlay that contains many boxes with dashed or dotted borders. See "Printer Storage Limitations" on page 335 and Appendix J, "Storage Summary" on page 341 for further details.

Shading: Another option available to you in your overlay definition is shading. The shading option has two different shade patterns (**STANDARD** and **SCREEN**) and each pattern has 32 shade types (percentage of shading). Legibility of text is influenced by the amount of shading required and the shade pattern chosen. Experiment with shaded boxes to determine which combination of pattern and type best meets your needs.

shading option **SHADE**

shade pattern Choose from:
STANDARD (Default)
SCREEN

Examples of **STANDARD** and **SCREEN** patterns in each available percentage of shade type are shown in Figure 211 on page 326 and Figure 212 on page 330.

shade type You can choose one of five available shades, by entering its name, or customize shading by entering a percentage. Choose from:

XLIGHT
LIGHT
MEDIUM (Default)
DARK
XDARK

n Percentage of shade, do not add the percent symbol (%).

Figure 37 illustrates the shade types available.

If you do not want a box shaded, leave out the subcommand entirely and follow the border type option with an end marker.

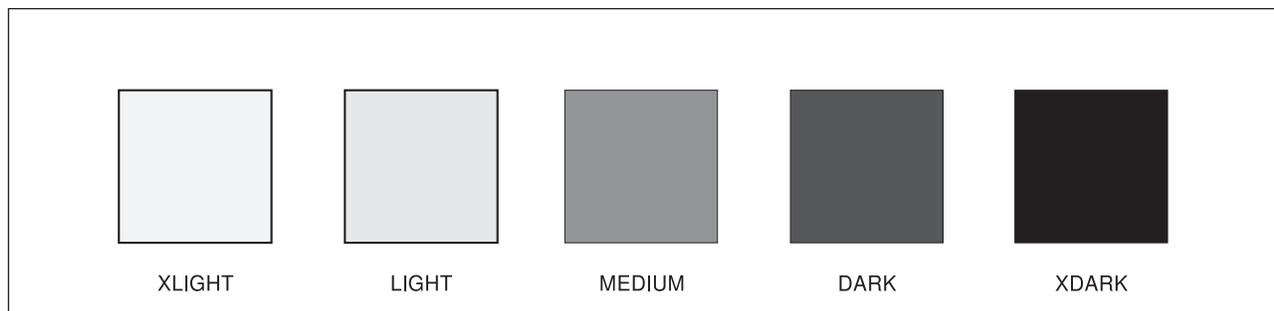


Figure 37. Named Shade Types for Boxes (**STANDARD**)

Box 3 has a light shade and a **MEDIUM** border.

You are ready to write the **POSITION/DRAWBOX** command pair for boxes 1 through 3 in Figure 32 on page 41. Do not bother with box 4 yet, because it contains text. We explain how to position text inside boxes in “Adding Text to Boxes (**DRAWBOX WITHTEXT**)” on page 100. When you have written the commands for Boxes 1, 2, and 3, compare them with Figure 38 on page 45.

```

-'DRAWING BOXES'
POSITION ABSOLUTE 0 ABSOLUTE 0;
DRAWBOX 7 IN 3 IN MEDIUM SOLID;

POSITION ABSOLUTE .25 IN ABSOLUTE .55 IN;
DRAWBOX .37 IN .37 IN BOLD SOLID;

POSITION ABSOLUTE 1 IN ABSOLUTE .55 IN;
DRAWBOX .75 IN 1 IN MEDIUM SOLID SHADE STANDARD LIGHT;

-'RECEIPT FRAME'

-'UNITS BOX'

-'FEES BOX'

```

Figure 38. **DRAWBOX** Commands for Overlay RCPT

When you are satisfied with your commands, include them in the sample overlay definition and send it to the OGL/370 processor. Compare the printout with Figure 39.

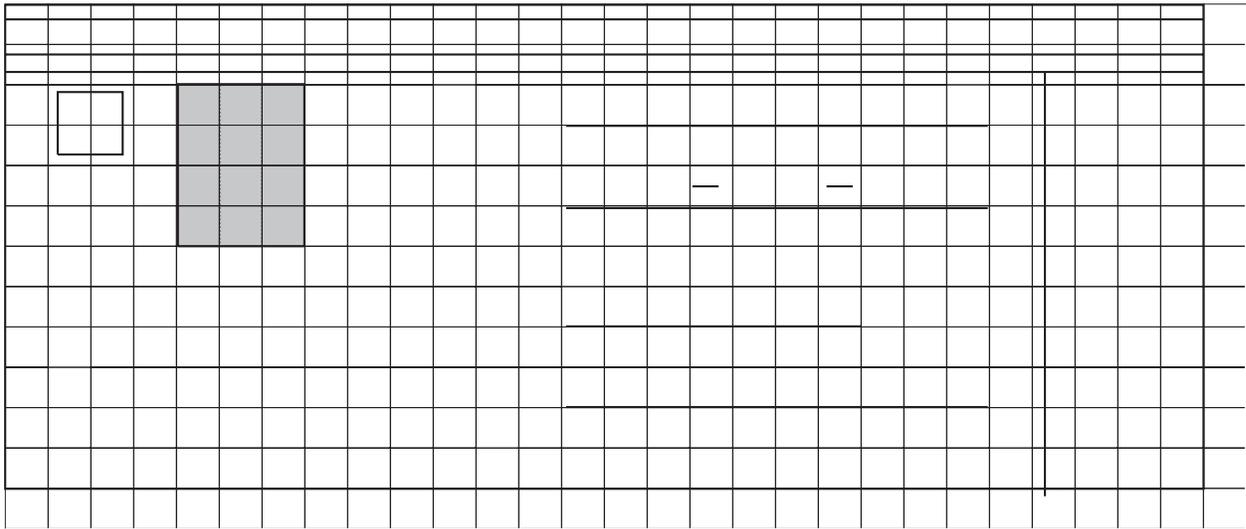


Figure 39. Printed Rules and Boxes

Relative Positioning (POSITION)

Refer to the “Cs” in Figure 41 on page 47. One way to position those “Cs” is to measure from the origin of the overlay. However, the “Cs” are very close to the “T”, a matter of a few pels. To determine the exact horizontal and vertical distances from the overlay origin to the “Cs”-pattern origin, you must:

1. Convert to pels the horizontal and vertical distances to the “T”-pattern origin.
2. Using pels, add or subtract the horizontal distances (right or left) from the “T” origin to the “C” origins. Then add the vertical distances.

An easier way is to use *relative positioning*.

You can also define a position relative to the coordinates of the last **POSITION** command. If the last **POSITION** command was for the origin of the “T”, you measure the horizontal and vertical distances from the “T” origin.

The parts of the **POSITION** command are as follows:

POSITION	LEFT	17 pels	DOWN	50 pels	;
command	origin	first	origin	second	end
word	option	coordinate	option	coordinate	marker

Figure 40. **POSITION** Command

command word

POSITION

POSITION This is the point from which you measure. When you choose any of the following entries, you are starting from the last **POSITION** command and measuring in the indicated direction. This is the horizontal direction, choose from:

LEFT
RIGHT
UP
DOWN

first coordinate Specify a number (*n*) and a unit of measurement:

n

IN Inches
MM Millimeters
PELS Pels

origin option This is the vertical direction from the last **POSITION** command, choose from:

LEFT
RIGHT
UP
DOWN

second coordinate

Specify a number (*n*) and a unit of measurement:

n

IN Inches
MM Millimeters
PELS Pels

end marker Always end a command with an end marker (;).

Before we show you an example of how relative positioning is used, let us look at the measurements and coordinates involved in placing the initials. We can use any of the standard units of measurement, but **PELS** is the most precise and easiest to use. Look at Figure 41.

A number 1 marks the coordinates of the last **POSITION** command (the origin of the “T”). To get to the position marked by number 2, move left 17 pels and then down 50 pels. Your new position command must give these directions and values. To place the origin of the “C” at position number 3, you measure from number 2. To get to the position marked by number 3, move right 74 pels and then down (or up) 0.

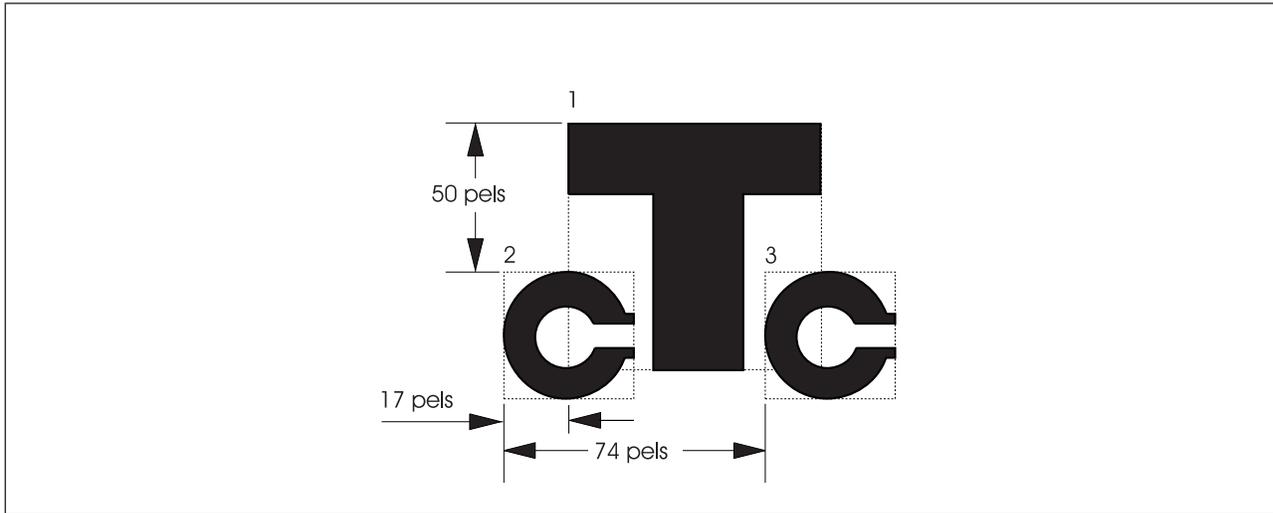


Figure 41. The Initials with Distances. (Not actual size.)

For the rest of the chapter, we refer to this new way of using the **POSITION** command as the *relative POSITION* command, and the one you were already familiar with we call the *absolute POSITION* command. They are really the same command, but the two terms prevent confusion.

Here are a few observations on the use of relative positioning:

- You can mix **ABSOLUTE** with **RIGHT**, **LEFT**, **DOWN**, or **UP** in the same **POSITION** command.
When **ABSOLUTE** is the first coordinate, it is the horizontal coordinate.
When **ABSOLUTE** is the second coordinate, it is the vertical coordinate.
- You must always specify two coordinates even if one is a 0 coordinate. For example, if you want to move right 1 inch and down 0 you can enter:

```
POSITION RIGHT 1 IN DOWN 0;
```

Be careful that you do not unintentionally specify an absolute coordinate. For example, the following command may not produce the same results as the previous one:

```
POSITION RIGHT 1 IN 0;
```

Because the default for the origin option of the second coordinate is **ABSOLUTE**, this command moves the position 1 inch to the right and to the top of the overlay.

- You may specify two horizontal or two vertical coordinates in the same relative **POSITION** command. If, for example, a box is 2 inches to the right of the current position and you want to set the new position to exactly 6 pels to the left of the box, you can specify the following:

```
POSITION RIGHT 2 IN LEFT 6 PELS;
```

This flexibility of the **POSITION** command can be useful in situations that require precise measurements.

You now can write the commands to position and place the “C”s. Write these commands, and compare the results with Figure 42.

```
DEFINE SMALLC PATTERN ENCODED (17 6)    -1
                                     .
                                     .
                                     (17 6);    140
POSITION LEFT 17 PELS DOWN 50 PELS;
PLACE PATTERN SMALLC 0;
POSITION RIGHT 74 PELS DOWN 0;
PLACE PATTERN SMALLC 0;
```

*Figure 42. A New Way to Use the **POSITION** Command*

You can now run the complete overlay definition, which should print an overlay like the one in Figure 14 on page 23 (except that yours includes the mask). Keep the **DRAWMASK** command in your definition for now. The mask is helpful if you need to make any final adjustments, which is the subject of the next section.

Fine-Tuning the Overlay

When all the parts of the overlay have been printed, you can decide if it looks the way it should. Perhaps a certain font is too large or too small. Perhaps a box or a rule is the wrong size, or the thickness or shading is not exactly what you want. These are simple changes. Most position changes are equally simple.

In some cases, however, you may want to move something just a few pels. To place the image SMALLC when pel measurements were critical, we used the relative **POSITION** command. We can do the same thing with other elements (rules, boxes, and segments). Look at the examples in Figure 43.

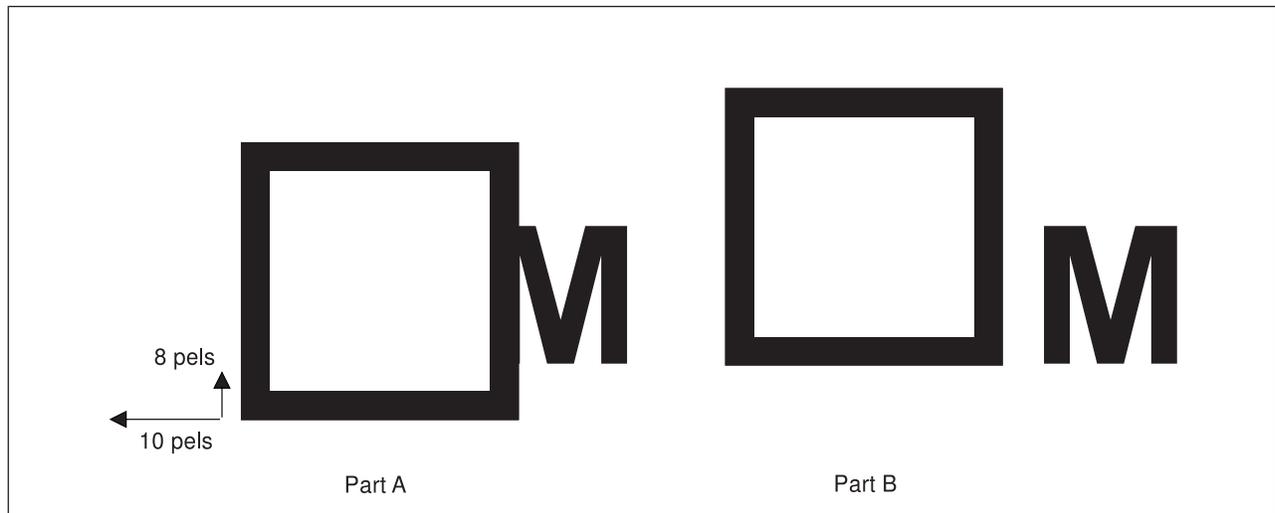


Figure 43. Fine-Tuning the Overlay

We wanted the bottom of the box aligned with the “M” and the edge of the box 6 pels from the “M”. Part A shows what was printed. The box was positioned 8 pels too low and 10 pels too far to the right. We can change the original **POSITION** command, but that can mean translating from one unit of measurement to another. An easier way is to follow the first **POSITION** command with a relative **POSITION** command.

Assume that you have defined and positioned the box in Figure 43, Part B as follows:

```
POSITION ABSOLUTE 6.25 IN ABSOLUTE 4.25 IN;  
DRAWBOX .2 IN .2 IN MEDIUM SOLID;
```

What command and at what origin positions the box 10 pels to the left and 8 pels up (as in Part B)?

```
POSITION ABSOLUTE 6.25 IN ABSOLUTE 4.25 IN;  
POSITION LEFT 10 PELS UP 8 PELS;  
DRAWBOX .2 IN .2 IN MEDIUM SOLID;
```

It is important that you can immediately follow one **POSITION** command with another **POSITION** command.

Note: You can position all boxes, text, rules, and graphics using the relative **POSITION** command. But to add, delete, or change the position of an element, you also have to change the **POSITION** command of every element following that change.

It is usually best to save the relative **POSITION** command for fine tuning the overlay. For example:

- After you have printed the overlay.
- When the position of one item depends on the position of another item (as was the case with the “T” and the “C”s in the image).

Cleaning Up

Once an overlay definition has been written and the overlay approved, make two important changes:

- Most likely you do not want the mask on the printed paper. Take out the **DRAWMASK** command.
- Change the **CONTROL** command so the overlay is stored and is thus available for use by application programs.

Using Defaults and Abbreviations

The complete definition for overlay “RCPT” appears in Appendix A, “Sample Overlay RCPT: Output Listing” on page 295. Using defaults and abbreviations saves time when typing commands.

The principle is that the most commonly used options (for example, a rule of **MEDIUM** thickness) are defaulted. **CONTROL** and **ORIENT** are fully defaulted. This means that even if you do not enter these commands, the defaults are chosen for you.

The keyword or its abbreviation can be used in the overlay definition. Appendix I, “OGL/370 Keywords” on page 337 presents a complete list of OGL/370 keywords and abbreviations.

Chapter 3. Drawing Circles and Paths

This chapter describes the **DRAWCIRCLE** and **DRAWPATH** commands.

Use the **DRAWCIRCLE** command to define circles and partial circles ($\frac{1}{4}$, $\frac{1}{2}$, or $\frac{3}{4}$ circles) for your overlay. Like boxes, circles can be shaded and can have text inside them.

You define arbitrary shapes using the **DRAWPATH** command. With this command, you define points on an overlay, that are joined together to form paths.

Drawing Circles

You may include whole or partial circles on your overlays. The circles illustrated in Figure 44 on page 54 are part of the sample overlay "RESULT", developed in Chapter 8, "Additional Features for Circles and Paths" on page 171.

There are two steps to drawing a circle:

1. Use a **POSITION** command to specify where the circle should be placed.
2. Use a **DRAWCIRCLE** command to describe the circle.

Drawing Circles

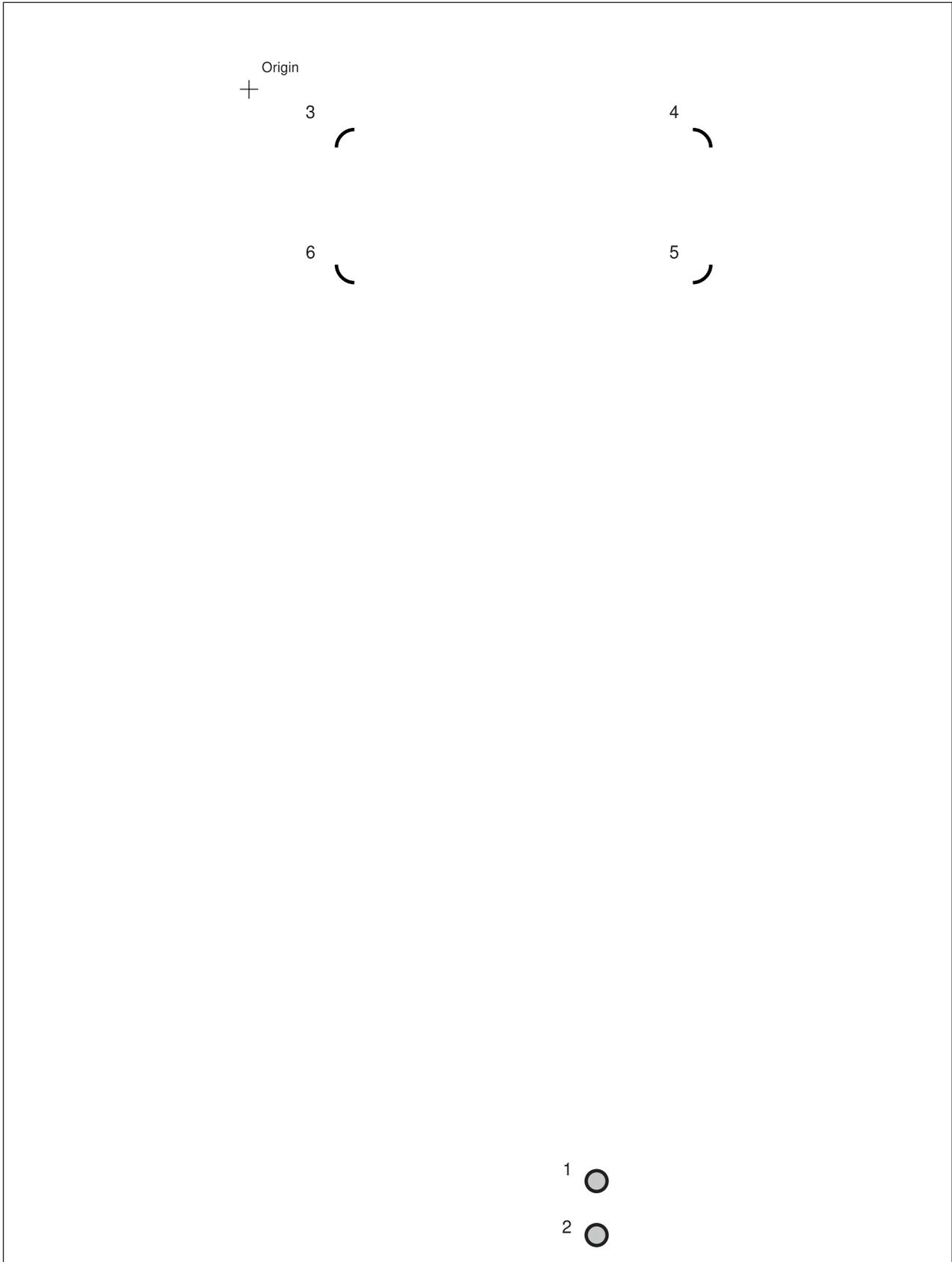


Figure 44. Extract from Overlay "RESULT" Used to Illustrate Circles. ("RESULT" is shown in full on Figure 157 on page 172)

Positioning the Circle (POSITION)

Define the position of a circle by specifying its center. You have already used the **POSITION** command to position rules and boxes. Use the **POSITION** command to position Circle 1 in Figure 44 on page 54.

POSITION	<u>ABSOLUTE</u>	2.44 in	<u>ABSOLUTE</u>	7.65 in	;
command word	origin option	first coordinate	origin option	second coordinate	end marker

Figure 45. **POSITION** (Circle) Command

Defining the Circle (DRAWCIRCLE)

Look at Circle 1 in Figure 44 on page 54. How would you describe this circle to someone who had not seen it? You would probably describe its size (radius), border (thickness and type), shape (whether it is a whole circle), and the shading. This is how you specify that information for the printer:

DRAWCIRCLE	0.06 in	10	<u>SOLID</u>	<u>WHOLE</u>
command word	circle radius	border thickness	border type	circle portion
SHADE	<u>STANDARD</u>	LIGHT	;	
shading option	shade pattern	shade type	end marker	

Figure 46. **DRAWCIRCLE** Command

command word

DRAWCIRCLE

circle radius The circle radius is measured from the circle center to the middle of the border. To define the radius, specify a number (*n*) and a unit measurement:

n

IN Inches
MM Millimeters
PELS Pels

Figure 47 on page 56 is an illustration of a circle with a radius of ½-inch and a MEDIUM (4 pels) border thickness.

The result is a circle with an overall width of twice the radius **plus** the thickness of one border.

Normally, the additional size is not critical. However, if you want to place a circle so it is almost, but not quite, touching another object on the overlay, the exact size is important.

Drawing Circles

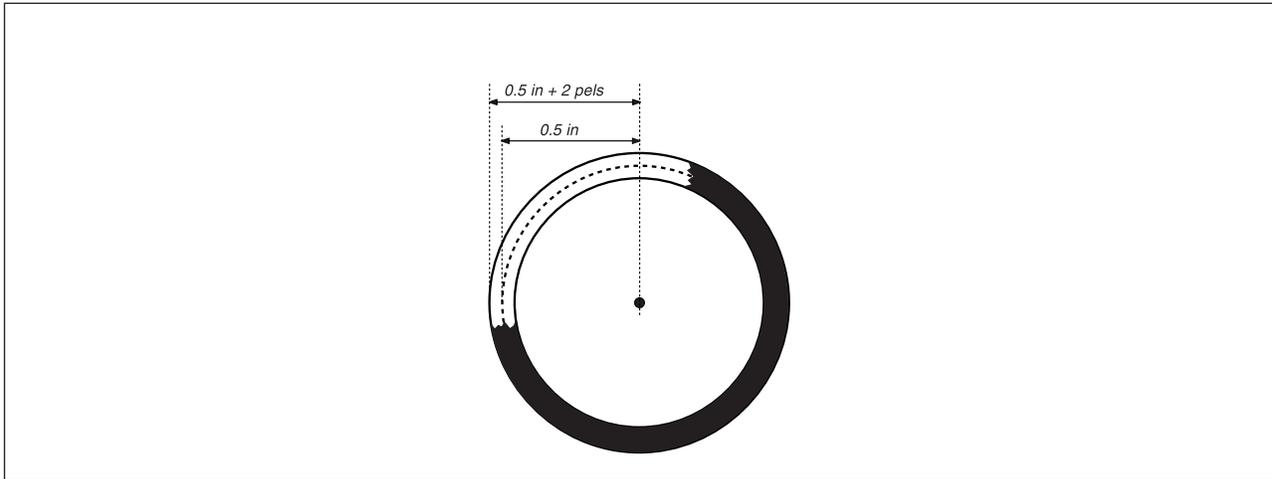


Figure 47. Circle Size

Border

border thickness

Describe the border thickness for a circle just as you do for a box or a rule. See Figure 29 on page 38 for thicknesses. Choose from:

LIGHT	2 pels thick.
MEDIUM	4 pels thick (default)
BOLD	6 pels thick
<i>n</i>	Thickness in pels, do not add the word “pels”.

Note: A border thickness of 0 creates an invisible circle, but you can still specify text and shading for it.

border type

The border types are the same as those in other commands. See Figure 30 on page 39 for types. Choose from:

SOLID	(Default)
DASHED	
DOTTED	

Shape: When defining a circle on your overlay, you can specify the type of circle you want drawn, a whole circle or a partial circle. You can specify a $\frac{1}{4}$, $\frac{1}{2}$, or $\frac{3}{4}$ -circle by using a combination of the appropriate circle identifiers. For example, a $\frac{3}{4}$ -circle can be drawn by specifying $\frac{3}{4}$, or by combining $\frac{1}{2}$ and $\frac{1}{4}$.

circle portion Define the circle or partial circle you want with the following commands:

WHOLE	A whole circle is drawn (default).
HALF	A $\frac{1}{2}$ -circle is drawn. Specify which $\frac{1}{2}$ of the circle you want drawn. See Figure 48 on page 57. Choose from: TOP BOTTOM LEFT RIGHT

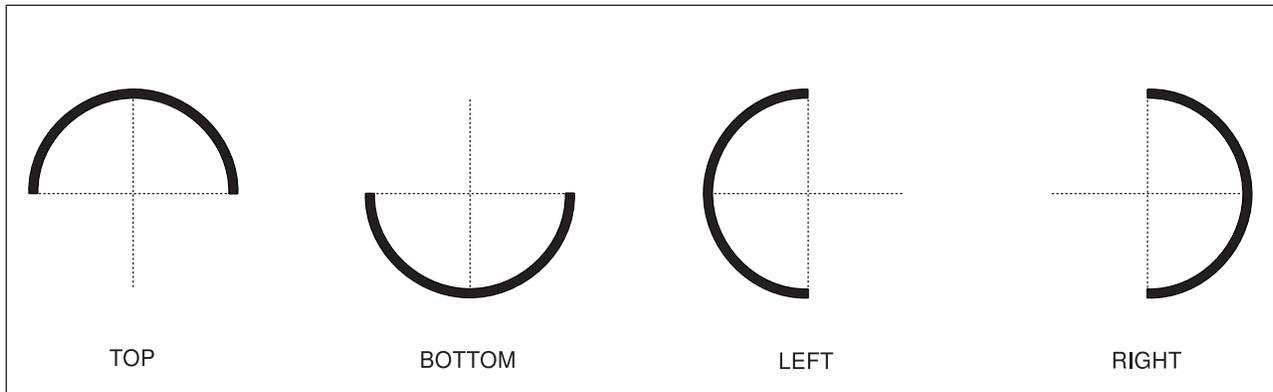


Figure 48. Half-Circles

QUARTER

A $\frac{1}{4}$ -circle is drawn. Specify which $\frac{1}{4}$ of the circle you want drawn. See Figure 49. More than one $\frac{1}{4}$ of a circle can be drawn for a specified circle position. Select one or a combination of the following:

- TOPLEFT**
- TOPRIGHT**
- BOTTOMLEFT**
- BOTTOMRIGHT**

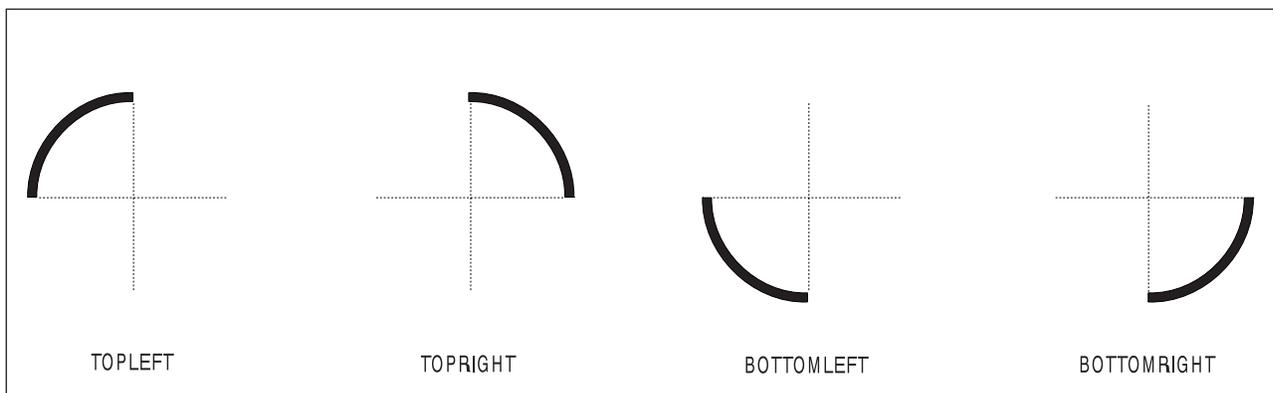


Figure 49. Quarter-Circles

Shading: You can specify shading for circles, the commands are the same as for box shading. There are two shade patterns and 32 shade types (percentage of shading).

shading option **SHADE**

shade pattern Choose from:
STANDARD (Default)
SCREEN

shade type Choose from:
XLIGHT
LIGHT
MEDIUM (Default)
DARK
XDARK

n Percentage of shading, do not add the percent symbol (%).

Positioning the Path (POSITION)

To define the position of a path, define its starting point. Recall that rules are positioned by defining the top-left corner. If a horizontal rule is 6 pels thick, the rule starts at the specified point and extends down from there by 6 pels. Paths, however, use center positioning. This means that if you define a path to be 6 pels thick, the path is centered on the starting point you specify. The line extends 3 pels to either side of this point. Figure 51 illustrates a rule and a path, each having a thickness of 6 pels. Both are positioned at the same place. However, the rule is 3 pels lower because top-left positioning was used.

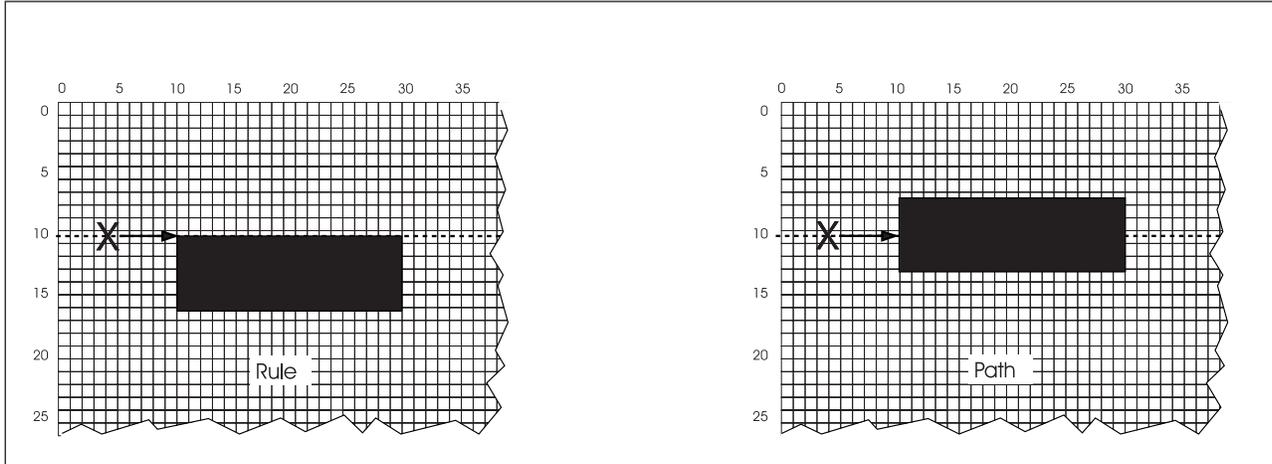


Figure 51. Positioning of a Rule and a Path. (Not actual size.)

The **POSITION** command for Path 1 in Figure 50 on page 58 is shown in Figure 52.

POSITION	<u>ABSOLUTE</u>	0.44 in	<u>ABSOLUTE</u>	1.94 in;
command	origin	first	origin	second
word	option	coordinate	option	coordinate

Figure 52. **POSITION** (Path) Command

Drawing Paths

Defining the Path (DRAWPATH)

Look at Path 1 in Figure 50 on page 58. How would you describe the path to someone who had not seen it? You would want to describe the shape of the path, and the type and thickness of the border. Define the shape of a path by specifying points on the overlay to be joined to form the path. Figure 53 shows the command used to define Path 1.

DRAWPATH	14	SOLID	CONNECTION	ROUNDED
command word	path thickness	path type	subcommand word	connection type
T0	ABSOLUTE	3.72 in	ABSOLUTE	1.94 in
T0	ABSOLUTE	3.72 in	ABSOLUTE	3.81 in
T0	ABSOLUTE	6.97 in	ABSOLUTE	3.81 in
T0	ABSOLUTE	6.97 in	ABSOLUTE	7.03 in
T0	ABSOLUTE	0.44 in	ABSOLUTE	7.03 in
subcommand word	origin option	first coordinate	origin option	second coordinate
CLOSE	;			
subcmd word	end marker			

Figure 53. **DRAWPATH** Command

command word

DRAWPATH

path thickness Choose from:

- LIGHT** 2 pels thick.
- MEDIUM** 4 pels thick (default).
- BOLD** 6 pels thick.
- n* Thickness in pels, do not add the word “pels”.

See Figure 29 on page 38 for thicknesses.

path type The path types are the same as border types for other commands. Choose from:

- SOLID** (Default)
- DASHED**
- DOTTED**

See Figure 30 on page 39 for types.

subcommand word

CONNECTION

connection type

Select the type for all connections in the path. Choose from:

- MITER** (Default)
- ROUNDED**

ROUNDED connections are described in “More Features of **DRAWPATH**” on page 179.

The connection type chosen here is used as the default for all connections in the path. However, on individual points, a connection type can be specified to override this default. Figure 54 illustrates the different connection types.

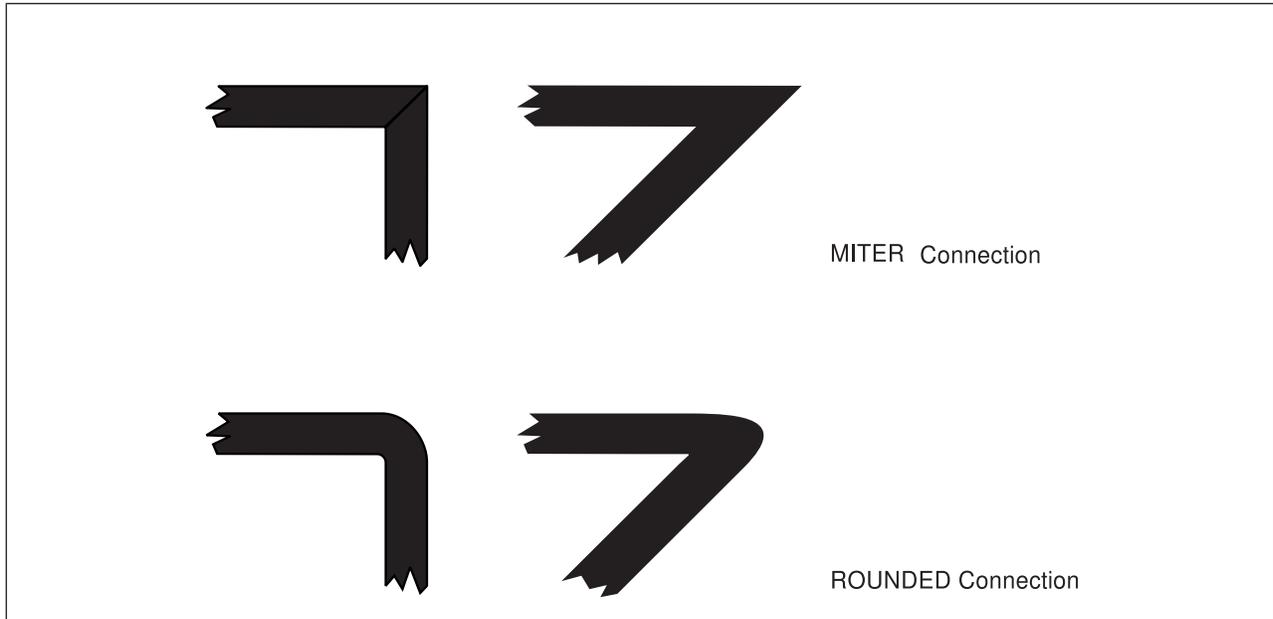


Figure 54. Path Connection Types

subcommand word

TO

origin option Specify whether you are going to use absolute or relative positioning. Choose from:

ABSOLUTE (Default)

LEFT

RIGHT

UP

DOWN

first coordinate Specify a number (*n*) and a unit of measurement:

n

IN Inches

MM Millimeters

PELS Pels

origin option Specify whether you are going to use absolute or relative positioning. Choose from:

ABSOLUTE (Default)

LEFT

RIGHT

UP

DOWN

second coordinate

Specify a number (*n*) and a unit of measurement:

n

IN Inches

MM Millimeters

PELS Pels

Drawing Paths

You can override the default connection type and specify a connection type for this point, using the **CLOSE** option. This is not illustrated in our example, however we describe it here.

subcommand word

CONNECTION

connection type

Select the type for the connection at this point on the path. Choose from:

MITER (Default)

ROUNDED

A sample of the different connection types is shown in Figure 54 on page 61. **ROUNDED** connections are described in “More Features of **DRAWPATH**” on page 179.

close option

Enter the following word:

CLOSE

to draw a line to the first point of the path—to close it.

Specify a connection type after the keyword **CLOSE**, as if it were a pair of coordinates.

Note: You can also close the path by giving the coordinates of the first point instead of using the **CLOSE** option.

end marker

Always end a command with an end marker (;).

The **DRAWPATH** command can be used to draw closed paths such as Path 1 in Figure 50 on page 58. **DRAWPATH** can also be used to draw lines. These lines can be horizontal, vertical, or diagonal. A line is an open path; it does not start and end at the same point on the overlay. An example of an open path is Path 2 in Figure 50 on page 58. The command used to draw this path is as follows:

DRAWPATH	<u>MEDIUM</u>	<u>SOLID</u>			
command word	path thickness	path type			
TO	DOWN	3.05 in	LEFT	0	;
subcmd word	origin option	first coordinate	origin option	second coordinate	end marker

Figure 55. **DRAWPATH** Command

command word

DRAWPATH

path thickness Choose from:

LIGHT 2 pels thick.

MEDIUM 4 pels thick (default).

BOLD 6 pels thick.

n Thickness in pels, do not add the word “pels”.

See Figure 29 on page 38 for the thicknesses.

path type

Choose from:

SOLID (Default)

DASHED

DOTTED

See Figure 30 on page 39 for the types.

subcommand word

TO

origin option Specify whether you are going to use absolute or relative positioning. Choose from:

ABOLUTE (Default)

LEFT

RIGHT

UP

DOWN

first coordinate Specify a number (*n*) and a unit of measurement:

n

IN Inches

MM Millimeters

PELS Pels

origin option Specify whether you are going to use absolute or relative positioning. Choose from:

ABOLUTE (Default)

LEFT

RIGHT

UP

DOWN

second coordinate

Specify a number (*n*) and a unit of measurement:

n

IN Inches

MM Millimeters

PELS Pels

end marker Always end a command with an end marker (;).

Note: There are storage considerations when designing an overlay that contains many paths. See “Printer Storage Limitations” on page 335 for further details.

Drawing Paths

Chapter 4. Adding Text

This chapter describes what you need to know about putting text in an overlay.

In OGL, you can put text in an overlay using the **SETTEXT** command or using the **WITHTEXT** subcommand in the **DRAWBOX**, **DRAWCIRCLE**, or **DRAWGRAPHIC** commands. Either way you go, the text is specified with the **LINE** subcommand in the commands mentioned above.

OGL/370 has limited text formatting capabilities. It only processes a single line of text at a time and does not flow text across lines.

Text Orientation and Positioning

You can place text in 0°, 90°, 180°, and 270° orientations, and you can also place text in Modern, Column, or Tate format.

You can position text relative to a box or a circle, and you can also position text anywhere on the overlay.

For more information on text orientation and positioning, see “Positioning and Spacing the Text” on page 76.

Terms Used to Describe Text

Text is made up of single-byte characters, double-byte characters, or both. Here is what we mean by single-byte characters and double-byte characters:

Single-byte characters (SBCS): Characters that use a single byte (eight bits) for each code point. Up to 256 code points can be defined by a single byte. Some examples of single-byte characters are the characters that make up the English, French, German, and Spanish languages.

Double-byte characters (DBCS): Characters that use two bytes (sixteen bits) for each code point. Up to 65,536 code points can be defined by two bytes. The Kanji language is one language that contains far more than 256 characters. Therefore, in order to use the Kanji language, double-byte fonts must be used.

We talk about the shift out and shift in code points as SOSI characters. (SOSI is the acronym for shift out and shift in.) Here is an explanation of the SOSI code points:

Shift Out (X'0E') A special code point that indicates that the text following it is double-byte text until a Shift In is encountered.

Shift In (X'0F') A special code point that indicates that the text is switching from double-byte text to single-byte text. OGL/370 only checks for this code point when processing double-byte text.

Note: In the text command examples in this manual, we use a < symbol to represent a SO code point and a > symbol to represent a SI code point.

SOSIINOSOSI Option of the CONTROL Command

The **CONTROL** command has a **SOSIINOSOSI** option. You can only specify this option once in an overlay.

When you specify **SOSI** in the **CONTROL** command, all double-byte text specified in the overlay must be delimited by SOSI code points.

When you specify **NOSOSI** in the **CONTROL** command, all double-byte text specified in the overlay *must not* be delimited by SOSI code points. With **NOSOSI**, you cannot mix single-byte text and double-byte text in a single set of quotation marks.

Specifying Text

The actual text is specified in the **LINE** subcommand of the **DRAWBOX**, **DRAWCIRCLE**, or **SETTEXT** commands. You use the **LINE** subcommand to specify the fonts, use of underlining, the text type, and the text as illustrated in Figure 56.

LINE	font1	NOUNDERLINE	CHAR	'DO '
	font2			'NOT'
	font1			' WRITE IN THE AREA BELOW';
LINE	font1	NOUNDERLINE	HEX	'C1C2C3'
line	font	underlining	text	text
subcmd	name	option	type	

Figure 56. **LINE** Subcommand for Single-Byte Text

You start the **LINE** subcommand with a font name that has been defined in a **FONT** command. You must enter a font name in each **LINE** subcommand before the text string. You may enter a different font name for each text segment. When printing text segments, OGL/370 uses the most recently specified font in the current **LINE** subcommand.

Note: Each operating system uses a different **FONT** command because each operating system has a different method of identifying font libraries. See “**FONT** Command (MVS)” on page 258, “**FONT** Command (VSE)” on page 262, and “**FONT** Command (VM)” on page 260 for more information.

If you want to underline text, you must specify underline before each text segment.

You have the choice of two text types: characters (**CHAR**) or hexadecimal (**HEX**) code points. You can specify any text in its character form or its hexadecimal form.

Notice that the text in Figure 56 has a blank specified after ‘DO’ and before ‘WRITE’ so that the spaces are included on the printed line; you can specify the blank before or after the word.

Parts of a Text Line

The text entered with the **LINE** subcommand consists of the following:

Text Segment Text enclosed in a single set of quotation marks.

Text String All the text specified in a single **LINE** subcommand. A text string contains one or more text segments.

```
LINE sf1 'This is the first text segment in this text string.'
      sf2 ' This is another text segment in this text string.'
      ' This is the last text segment in this text string.'

LINE sf1 'This is another text string.'
```

Figure 57. Examples of Text Segments and Text Strings in the **LINE** subcommand

Specifying Single-Byte Text

If you only use single-byte text, you need not choose between **SOSI** and **NOSOSI**, OGL/370 defaults to **SOSI**. However, if your single-byte font contains characters that are assigned to the X'0E' or X'0F' code point and you use these characters in your overlay, you should specify the **NOSOSI** option.

If your overlays only use single-byte text, you can skip the discussion on double-byte text.

Specifying Double-Byte Text with the NOSOSI Option

When your overlay has double-byte text, you must specify a double-byte font in the **LINE** subcommand before you specify the double-byte text, and that is the only difference between specifying double-byte and single-byte text. Remember, with the **NOSOSI** option, you must not delimit the double-byte text with the **SOSI** characters. Figure 58 illustrates the **LINE** subcommand for double-byte text.

In Figure 58:

- **NOSOSI** is specified in the **CONTROL** command.
- Single-byte text is specified in one text segment and double-byte text is specified in the other two text segments.
- A single-byte font (sbf) is specified before the text segment containing single-byte text, and a double-byte font (dbf) is specified before the first text segment containing double-byte text. That 'dbf' also applies to the second text segment containing double-byte text.

```
CONTROL NOSOSI;
:
LINE   sbf   NOUNDERLINE  CHAR  'ssss ssss' -'segment 1 - SBCS text'
        dbf                'dddd dddd' -'segment 2 - DBCS text'
                                'dddd dddd' -'segment 3 - DBCS text'
```

line subcmd	font name	underlining option	text type
	sbf		'ssss ssss'
	dbf		'dddd dddd'
			'dddd dddd'

Where:

segment 1: "ssss ssss" is printed with font sbf
segment 2: "dddd dddd" is printed with font dbf
segment 3: "dddd dddd" is printed with font dbf

Figure 58. Example of **LINE** Subcommand for Single-Byte and Double-Byte Text with **NOSOSI**

Specifying Double-Byte Text with the SOSI Option

When you use the **SOSI** option, you can have single-byte text segments, double-byte text segments, and mixed text segments in the same line of text. You can also specify a single-byte and a double-byte font before the text segments and because of the **SOSI** delimiters, OGL can determine which font to use. The fonts can be specified together or one at a time in the **LINE** subcommand.

Within a **LINE** subcommand, OGL uses the most recently specified single-byte font and the most recently specified double-byte font for the single-byte and double-byte text. Remember, with the **SOSI** option, you must delimit the double-byte text with the **SOSI** characters. In addition to the font selection, use of underlining, the text type, and the actual text, you can also specify whether or not a character space is to appear in the positions held by the **SOSI** characters.

For an explanation of the **SOSI** and **SOSI2** option, see pages 215, 227, and 289 in Part 4, "Reference" on page 189.

Specifying Text

Examples Specifying Double-Byte Text with the SOSI Option

In Figure 59:

- **SOSI** is specified in the **CONTROL** command.
- Single-byte and double-byte text are used both separately and together (mixed text).
- A single byte font (sfont) and a double byte font (dfont) are specified together at the beginning of the **LINE** subcommand.

```
CONTROL SOSI;      (Note: This is the default.)
:
DRAWBOX
:
LINE dfont sfont  '<dddd>'      -'segment 1 - DBCS text only'
                  'sssss'      -'segment 2 - SBCS text only'
                  'aaa<dddddd>b' -'segment 3 - Mixed text'
```

Where:

- segment 1: "dddd" is printed with font dfont
- segment 2: "sssss" is printed with sfont
- segment 3: "aaa" and "b" are printed with sfont
"dddddd" is printed with dfont

Figure 59. Example Using Double-Byte Characters and the **SOSI** Option

In Figure 60 on page 69:

- **SOSI** is specified in the **CONTROL** command.
- Single-byte and double-byte text are used both separately and together (mixed text).
- Multiple single-byte fonts (sfont1 and sfont2) and multiple double-byte fonts (dfont1 and dfont2) are used in the line.

```

CONTROL SOSI;      (Note: This is the default.)
:
DRAWCIRCLE
:
LINE dfont1 sfont1 '<dddd>'      -'segment 1 - DBCS text only'
                   'sssss'      -'segment 2 - SBCS text only'
                   'aaa<dddddd>b' -'segment 3 - Mixed text'
                   sfont2 'sssss' -'segment 4 - SBCS text only'
                   dfont2 '<dd>'  -'segment 5 - DBCS text only'
                   'aaa<dd>b'    -'segment 6 - Mixed text'

```

Where:

segment 1: "dddd" is printed with font dfont1
segment 2: "sssss" is printed with sfont1
segment 3: "aaa" and "b" are printed with sfont1
"dddddd" is printed with dfont1
segment 4: "sssss" is printed with sfont2
segment 5: "dd" is printed with font dfont2
segment 6: "aaa" and "b" are printed with sfont2
"dd" is printed with dfont2

Figure 60. Example Using Double-Byte Characters and the **SOSI** Option

Using Double-Byte Text in Hexadecimal with the SOSI Option: You can specify text in hexadecimal with the **SOSI** option, but you must surround the text with the code points for the **SOSI** characters. However, you can not specify mixed text in hexadecimal.

In Figure 61 on page 70:

- **SOSI** is specified in the **CONTROL** command.
- Double-byte text segment is specified using the **HEX** option.

Text Margins in Boxes and Circles

```
CONTROL SOSI;      (Note: This is the default.)
:
DRAWBOX
:
LINE dfont sfont hex '0E42C142C20F' -'segment 1 - DBCS (in hex)'
                  'sssss'          -'segment 2 - SBCS text'
                  hex 'C1C2'        -'segment 3 - SBCS (in hex)'
```

Where:

- segment 1: The characters assigned to code points x'42C1' and x'42C2' are printed with dfont. The 0E and 0F are discarded.
- segment 2: "sssss" is printed with sfont
- segment 3: the characters assigned to code points X'C1' and X'C2' will be printed with sfont.

Notes:

1. OGL knows that segment 1 is to be printed with the DBCS font because the text begins with X'0E' and ends with X'0F'.
2. **HEX** text does not have to be entered in uppercase.

Figure 61. Example Using Double-Byte **HEX** Characters and the **SOSI** Option

Text Margins in Boxes and Circles

This section describes and illustrates the way OGL/370 places text in boxes and circles.

Each **WITHTEXT** subcommand can contain one or more text strings. The group of text strings specified in one **WITHTEXT** subcommand is called a *text block*.

The size of a text block depends on the fonts, line spacing, and text being used.

When OGL/370 is placing a text block in a box or circle, it places the block with reference to the inside text margin. There are two types of text margins: **WITHTEXT** and **ROUNDED** (default).

text margins

ROUNDED The rounded text margin is exactly one border thickness inside the border of the box or circle.

Note: For a box, if the length of a rounded corner is greater than 1½ times the border thickness, the margin is also rounded. If the length is less than or equal to that, the margin is square and is handled exactly the same as in a box with only square corners.

SQUARE The square text margin inside a box with some rounded corners is the largest rectangle that fits inside the **ROUNDED** text margin.

The square text margin inside a circle is the largest rectangle that fits inside the **ROUNDED** text margin.

Text Margins in Boxes

The **DRAWBOX** command is used to create boxes with square corners, rounded corners, or a combination of both.

Boxes with Only Square Corners

The text margin for a box with only square corners is a rectangle located exactly one border thickness inside the box border. Thus, text margin positioning is dependent on the border thickness, which is selected for each box.

Notes:

1. The text margin for a box with only square corners is the same whether you select **SQUARE** or **ROUNDED** text margins in the **SETUNITS** command.
2. Text margins for boxes with dotted or dashed borders are the same as text margins for boxes with solid borders.

Boxes with Some Rounded Corners

Figure 62 illustrates **ROUNDED** and **SQUARE** margins for a box with all corners rounded.

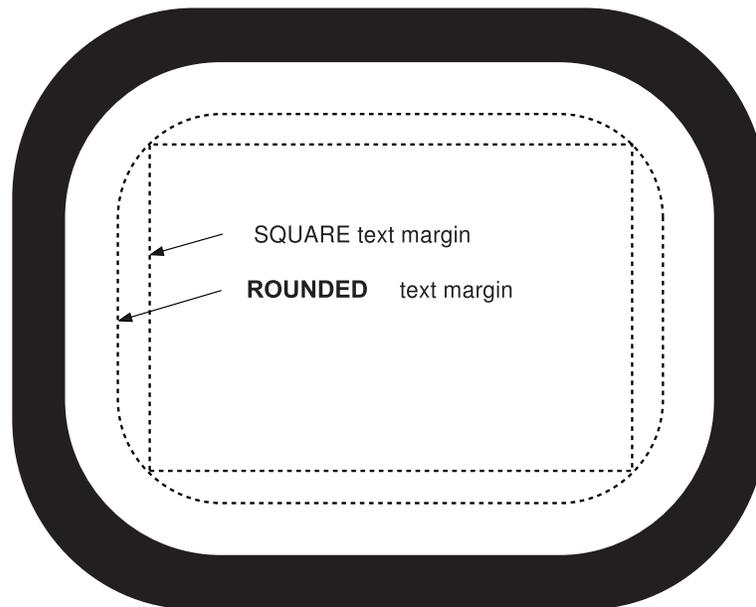


Figure 62. **DRAWBOX** Text Margins for a Box with All Corners Rounded

Figure 63 on page 72 illustrates margins used when only one corner is rounded (only one of the four possible cases is shown).

Text Margins in Boxes and Circles

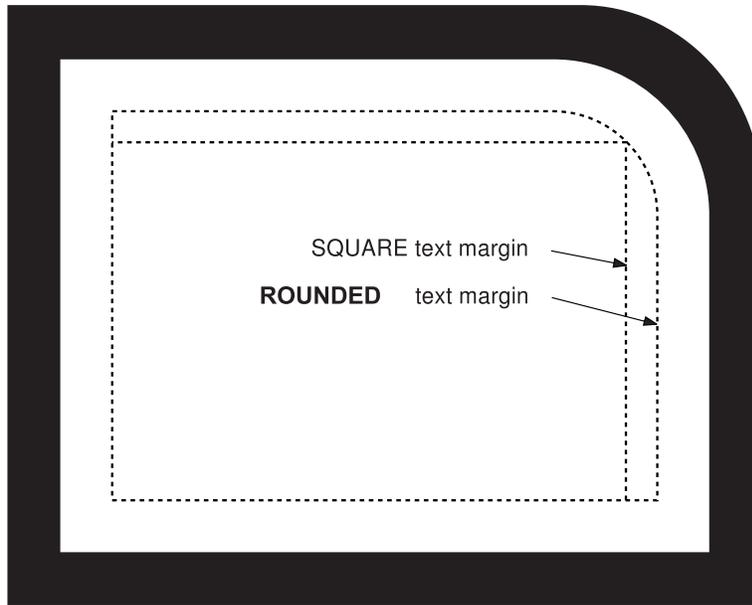


Figure 63. **DRAWBOX** Text Margins for a Box with One Rounded Corner

Figure 64 illustrates margins used when two adjacent corners are rounded (only one of the four possible cases is shown).

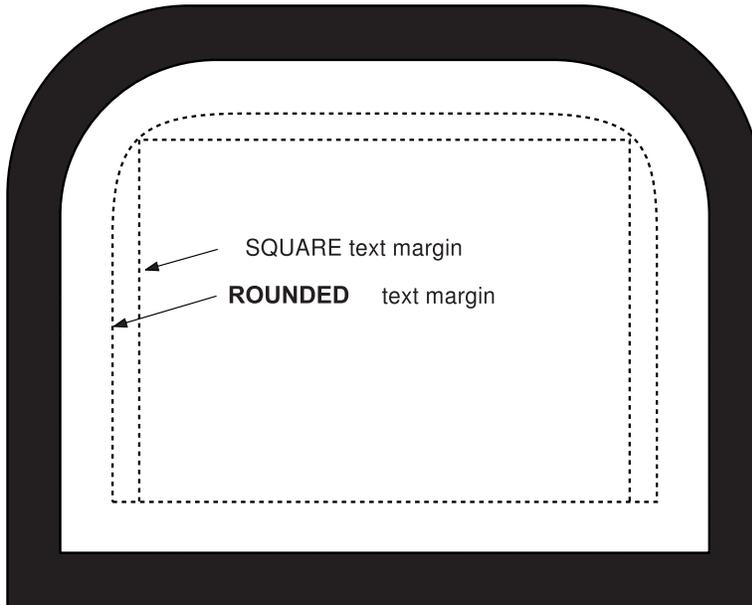


Figure 64. **DRAWBOX** Text Margins for a Box with Two Adjacent Rounded Corners

Figure 65 on page 73 illustrates margins used when two opposite corners are rounded (only one of the two possible cases is shown).

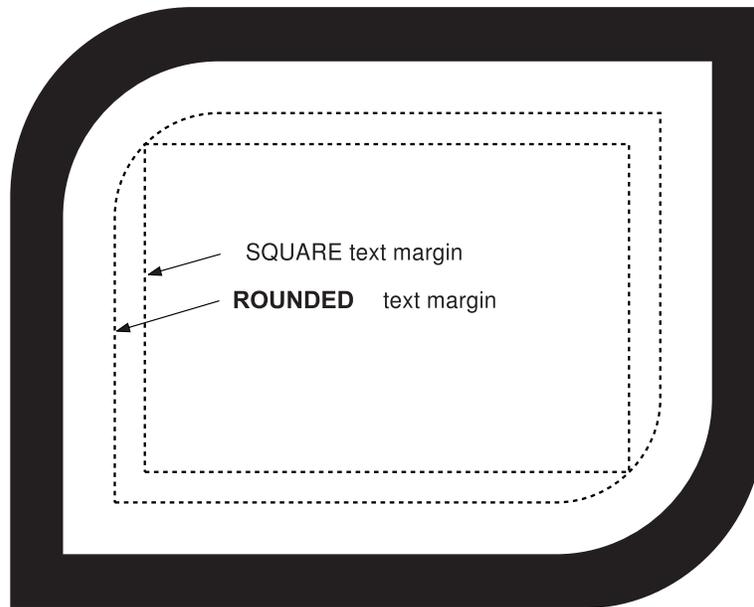


Figure 65. **DRAWBOX** Text Margins for a Box with Two Opposite Rounded Corners

Figure 66 illustrates margins used when three corners are rounded (only one of the four possible cases is shown).

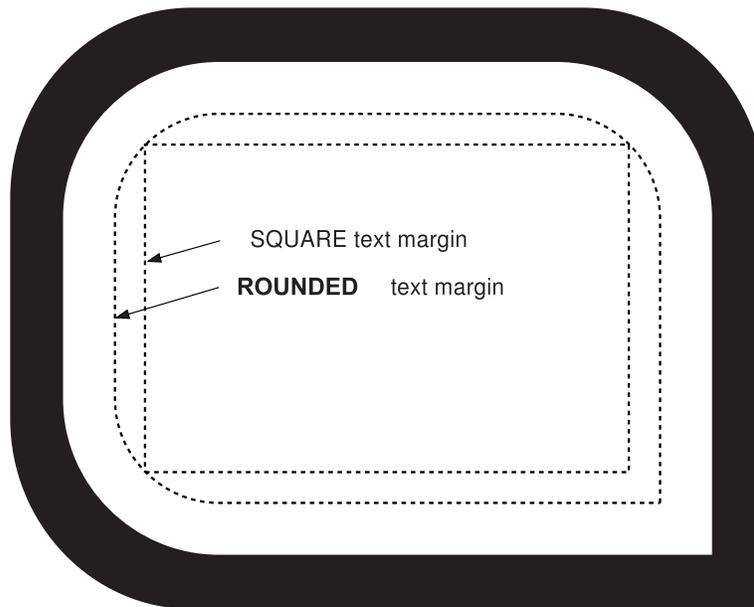


Figure 66. **DRAWBOX** Text Margins for a Box with Three Rounded Corners

Text Margins in Circles

For the purposes of text placement, partial circles are treated as whole circles, with text margins the same as those for a whole circle.

See Figure 67 on page 74 for examples of ROUNDED and **SQUARE** text margins in a whole circle.

Text Margins in Boxes and Circles

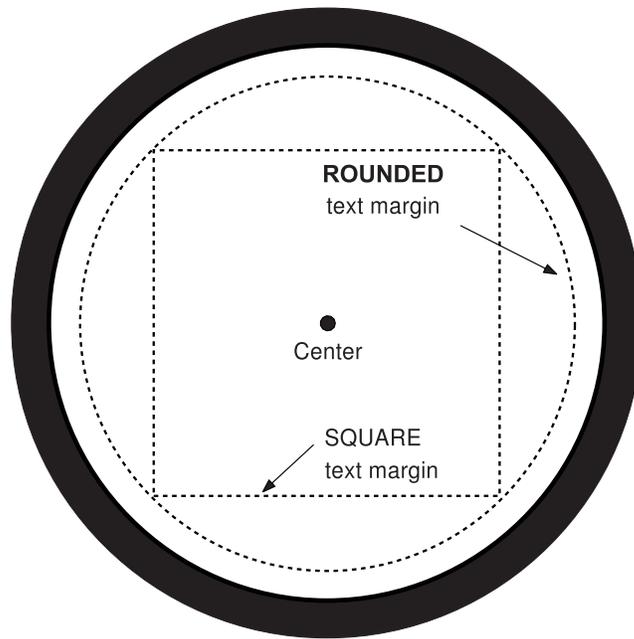


Figure 67. Text Margins in a **WHOLE** Circle

Figure 68 illustrates round and square margins for a $\frac{3}{4}$ circle.

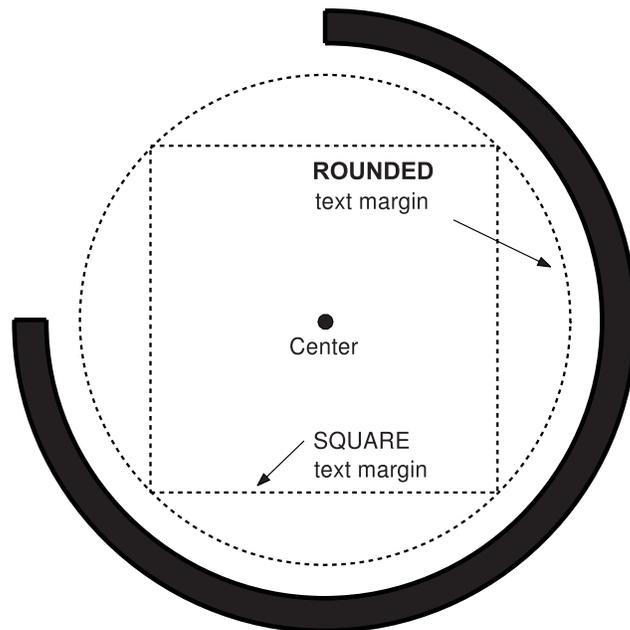


Figure 68. Text Margins in a $\frac{3}{4}$ Circle

Figure 69 on page 75 illustrates round and square margins for a $\frac{1}{2}$ circle.

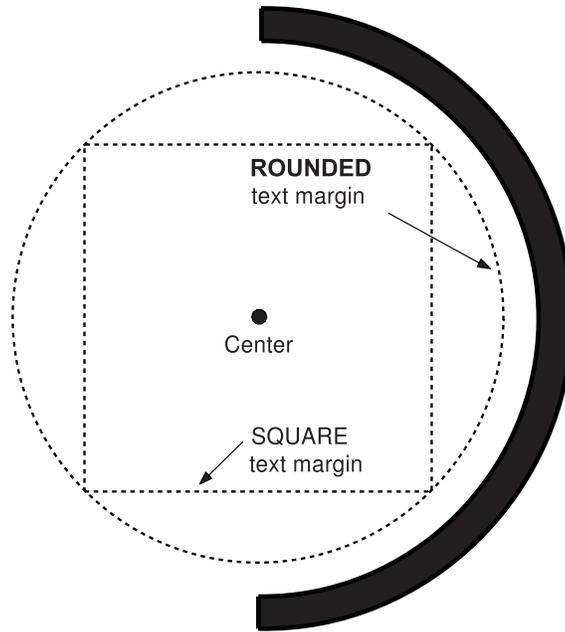


Figure 69. Text Margins in a 1/2 Circle

Figure 70 illustrates round and square margins for a 1/4 circle.

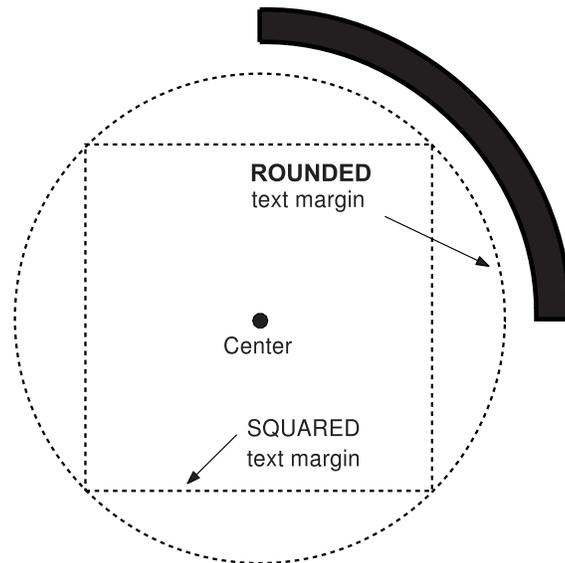


Figure 70. Text Margins in a 1/4 Circle

Positioning and Spacing the Text

Positioning and Spacing the Text

This section describes the format, orientation, and placement options available when defining text.

Text Format

You select the way text strings are arranged within a block by choosing one of three formats:

MODERN

This paragraph is written in the **MODERN** format. The characters are read from left to right, and the text strings are read from top to bottom (default).

The height of a text block is determined by the number of text strings in the block and the height of each text string. The height of a text string is the height of the largest font used in that text string.

For the horizontal positions described here (**LEFT**, **CENTER** default, and **RIGHT**), the width of a text block is determined by the length of the longest text string in the block. Other horizontal options are described in "Text Placement" on page 86.

COLUMN

The characters are read from top to bottom, and the text strings are read from left to right.

T L
W I
O N
E
S

The width of a text block is determined by the number of text strings in the block, and the width of each text string. The width of a text string is the width of the largest font used in that text string.

For the vertical positions described here (**TOP**, **CENTER** default, and **BOTTOM**), the height of a text block is determined by the length of the longest text string in the block. Other vertical options are described in "Text Placement" on page 86.

TATE

The characters are read from top to bottom, and the text strings are read from right to left. This is the traditional way of writing the characters of many Asian languages.

L T
I W
N O
E
S

The width and height of a **TATE** text block is determined in the same way as a **COLUMN** text block.

A text block can have one of nine positions. You select one of these by choosing a vertical position for the block (**TOP**, **CENTER**, or **BOTTOM**) plus a horizontal position (**LEFT**, **CENTER**, or **RIGHT**).

Inside any box or circle, OGL/370 defines a margin that governs the placement of text blocks. The width of the margin is equal to the thickness of the box or circle border. For example, if you select **LEFT** as the horizontal placement option, the left side of the block is positioned against the left margin; that is, one border thickness away from the inner edge of the left border. **RIGHT**, **TOP**, and **BOTTOM** work in the same way, as shown in Figure 71 on page 77, Figure 72 on page 77, and Figure 73 on page 78.

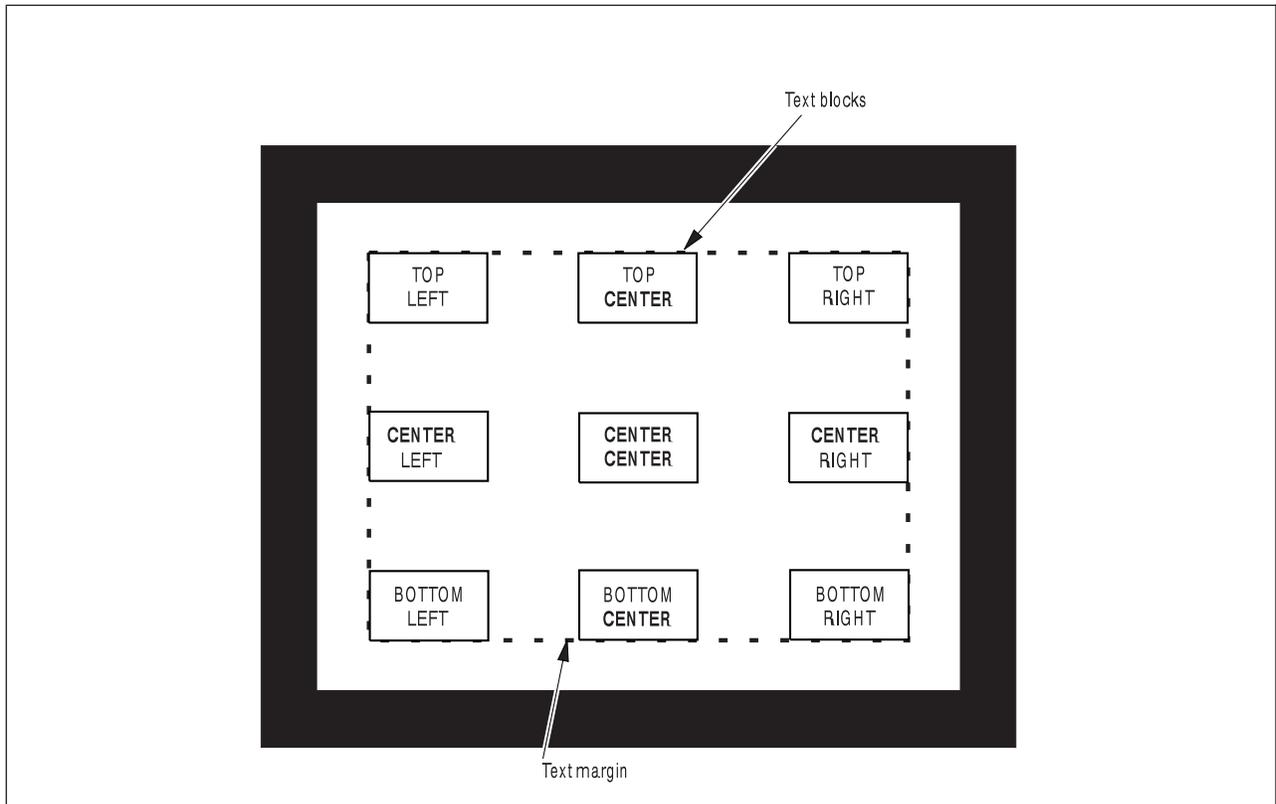


Figure 71. Text Positioning Options in a Box

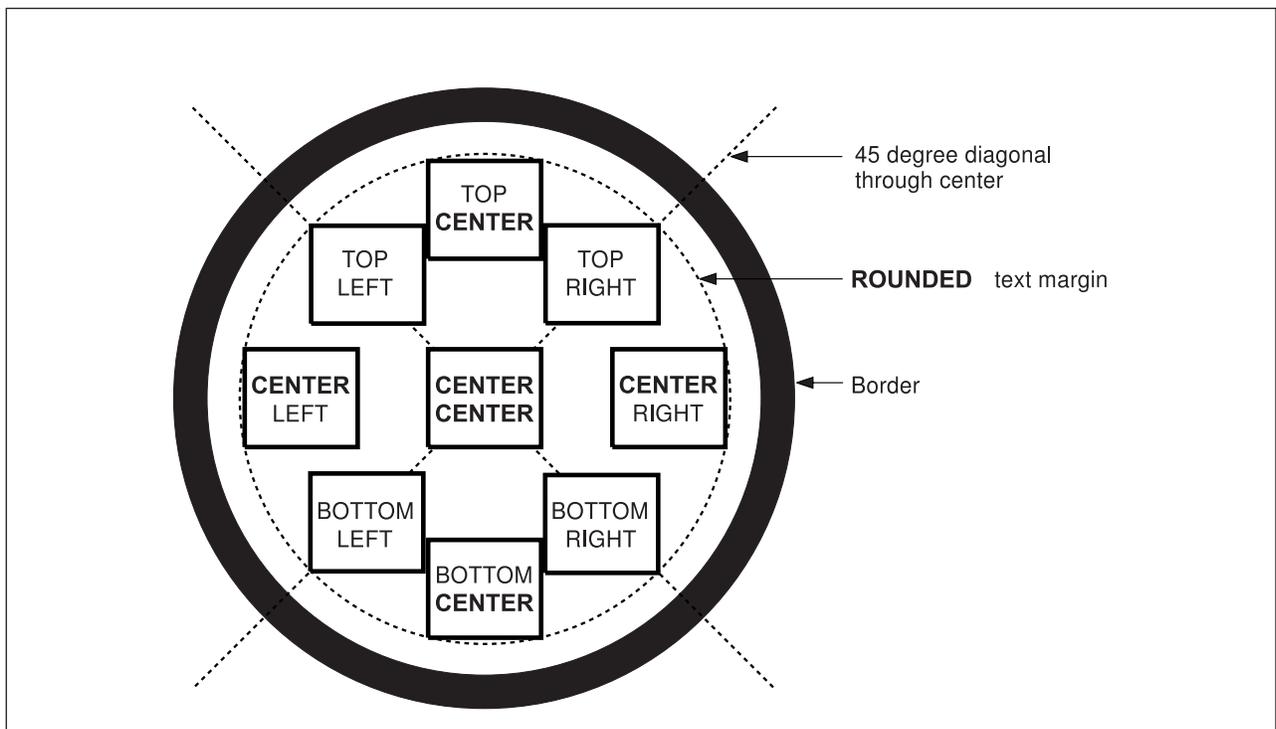


Figure 72. Text Positioning Options in a Circle — ROUNDED Margin

Positioning and Spacing the Text

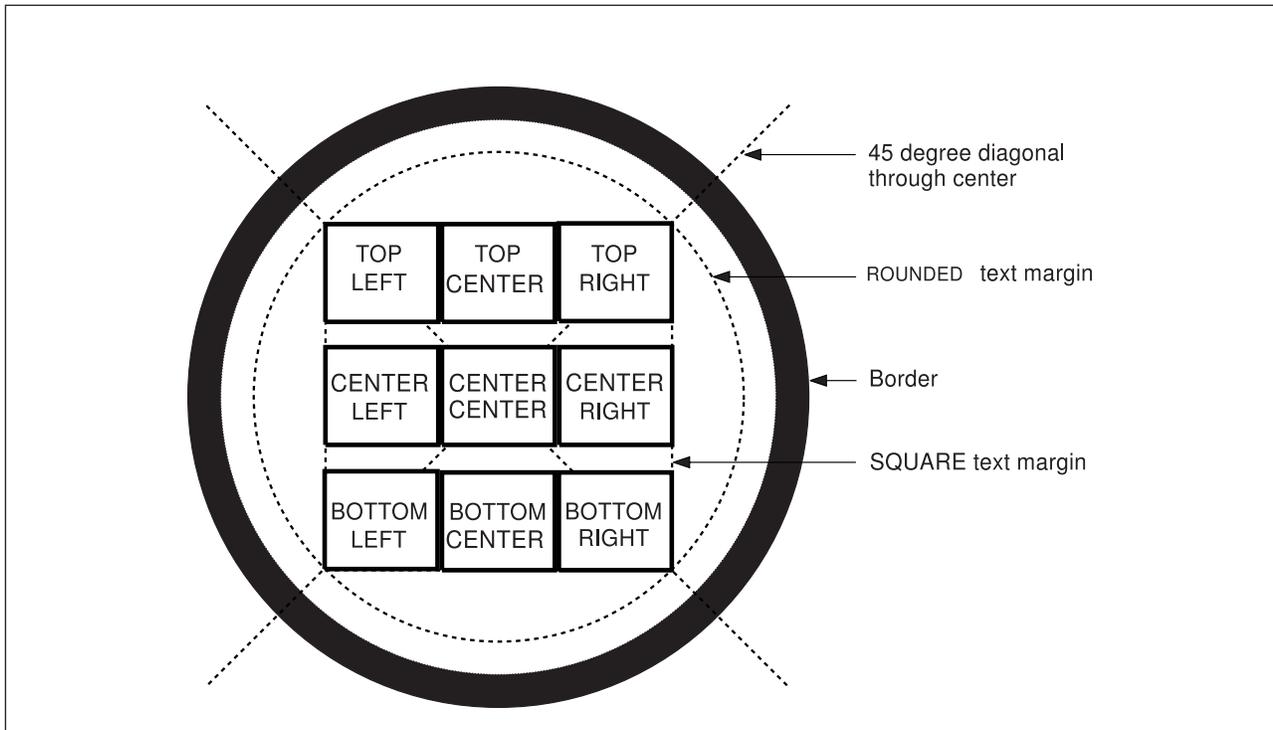


Figure 73. Text Positioning Options in a Circle — **SQUARE** Margin

Figure 74 on page 79 and Figure 75 on page 80 illustrate text-placement combinations for **MODERN** format.

Positioning and Spacing the Text

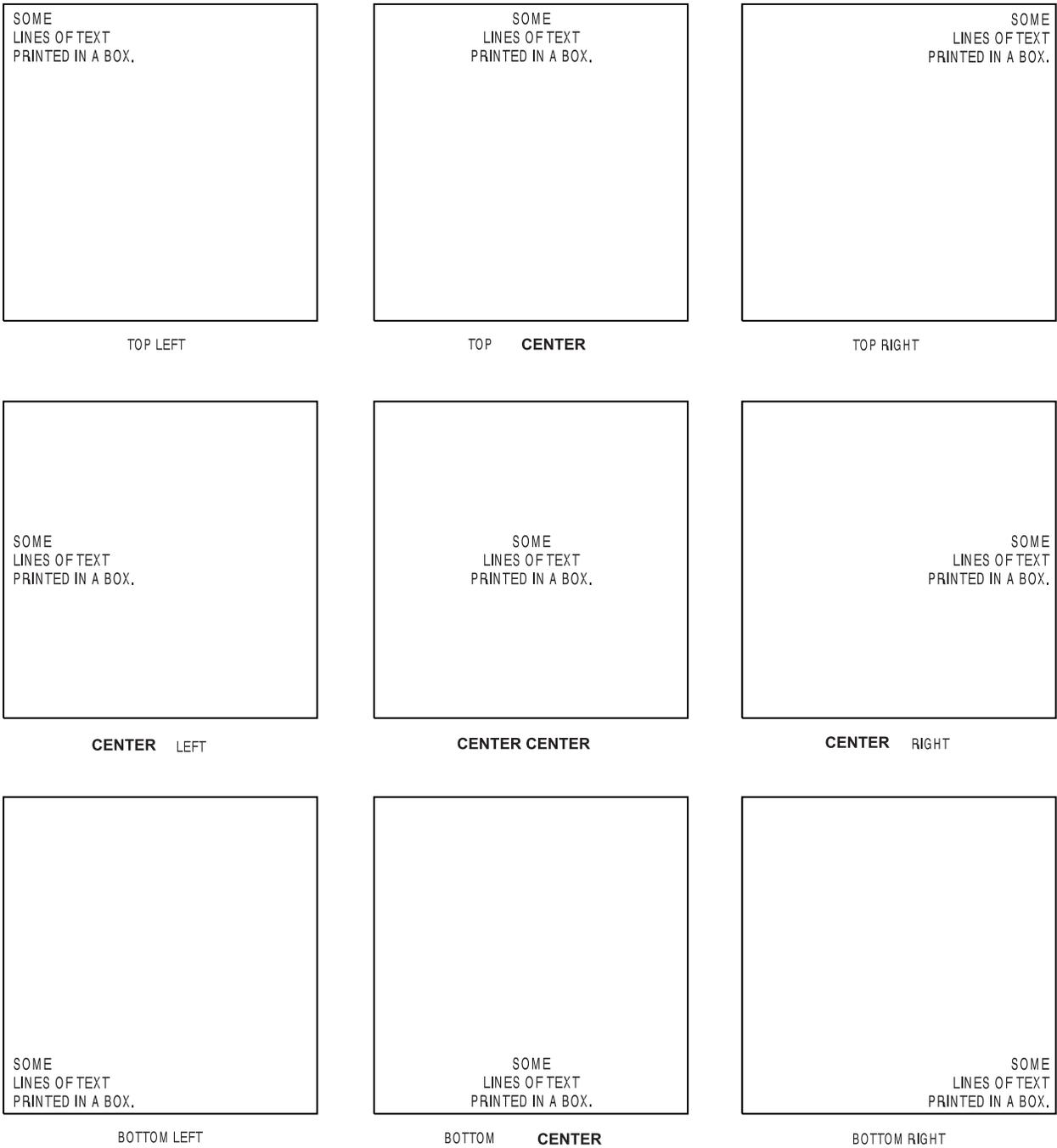


Figure 74. Placement of Text Written in **MODERN** Format. This example is approximate.

Positioning and Spacing the Text

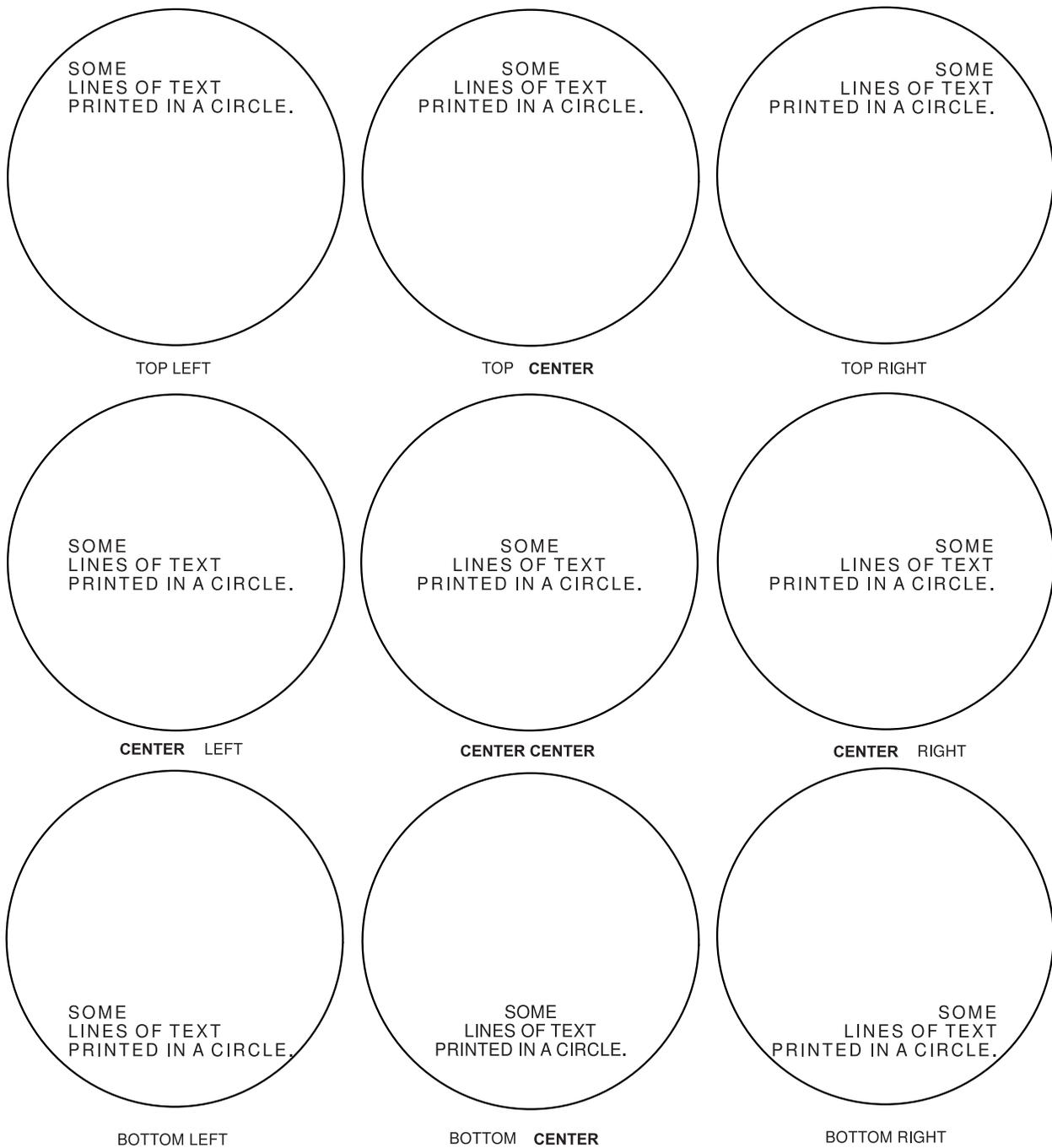


Figure 75. Placement of Text Written in **MODERN** Format in a **SQUARE** Text Margin. This example is approximate.

Figure 76 on page 81 and Figure 77 on page 82 illustrate text-placement combinations for **COLUMN** format.

Positioning and Spacing the Text

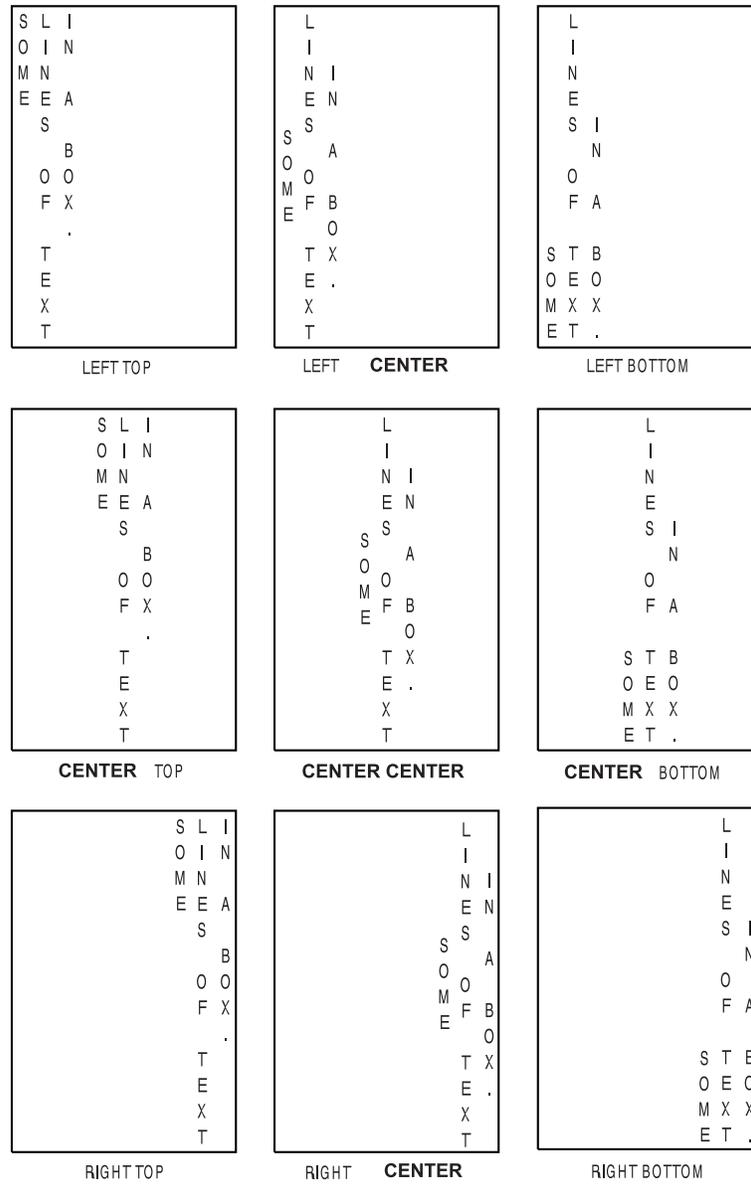


Figure 76. Placement of Text Written in **COLUMN** Format. This example is approximate.

Positioning and Spacing the Text

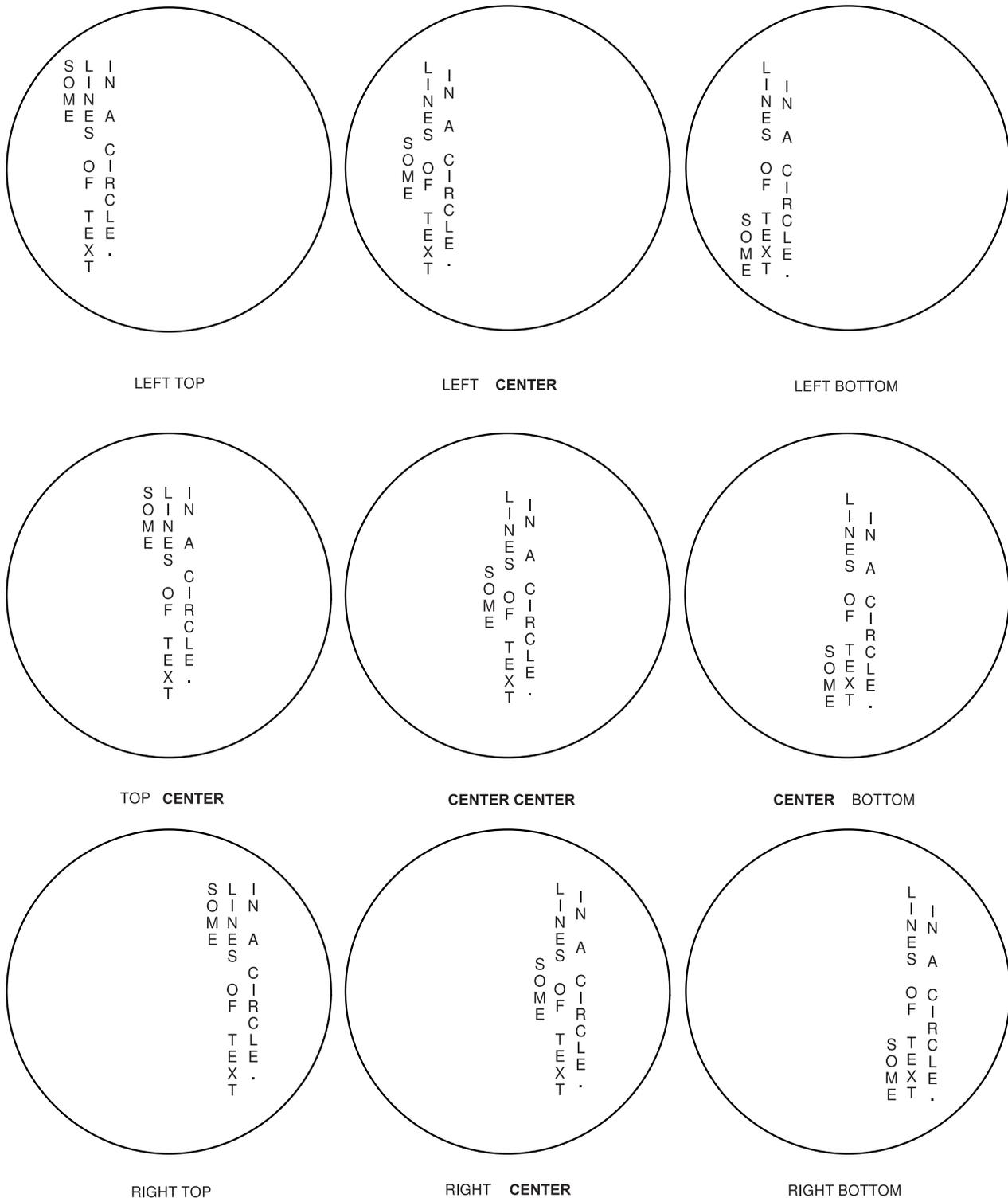


Figure 77. Placement of Text Written in **COLUMN** Format in a **SQUARE** Text Margin. This example is approximate.

Figure 78 on page 83 and Figure 79 on page 84 illustrate text-placement combinations for **TATE** format.

Positioning and Spacing the Text

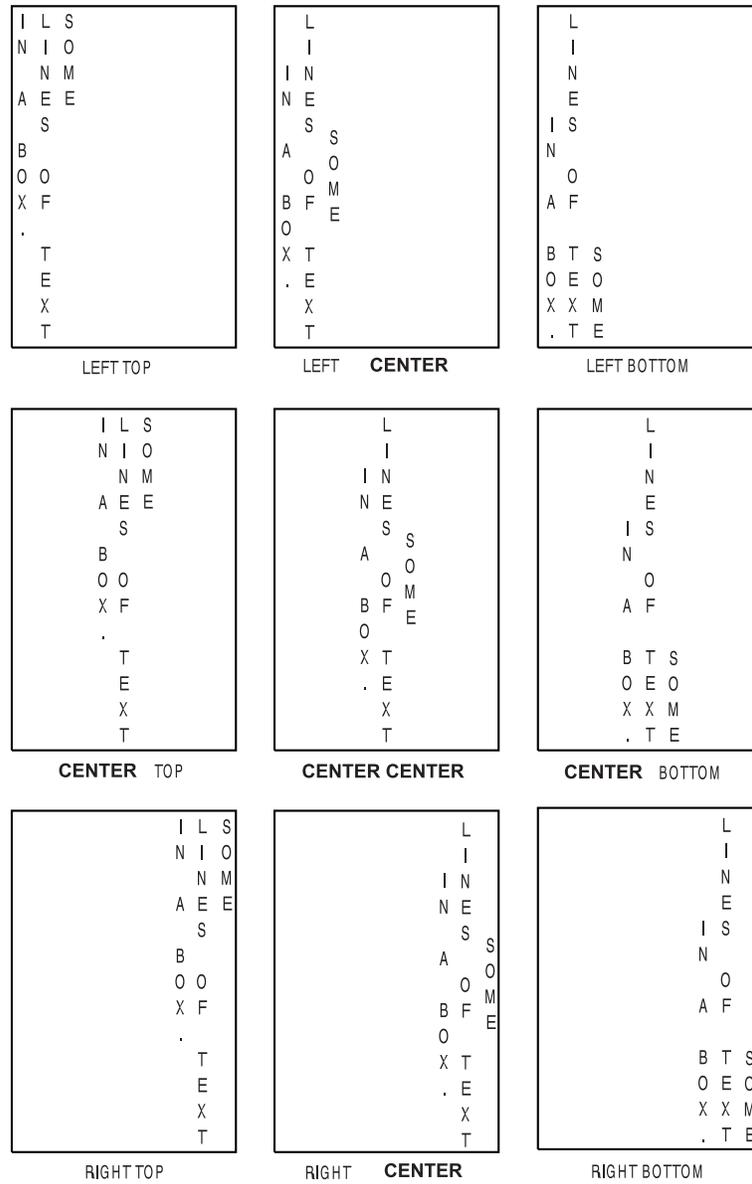


Figure 78. Placement of Text Written in **TATE** Format. This example is approximate.

Positioning and Spacing the Text

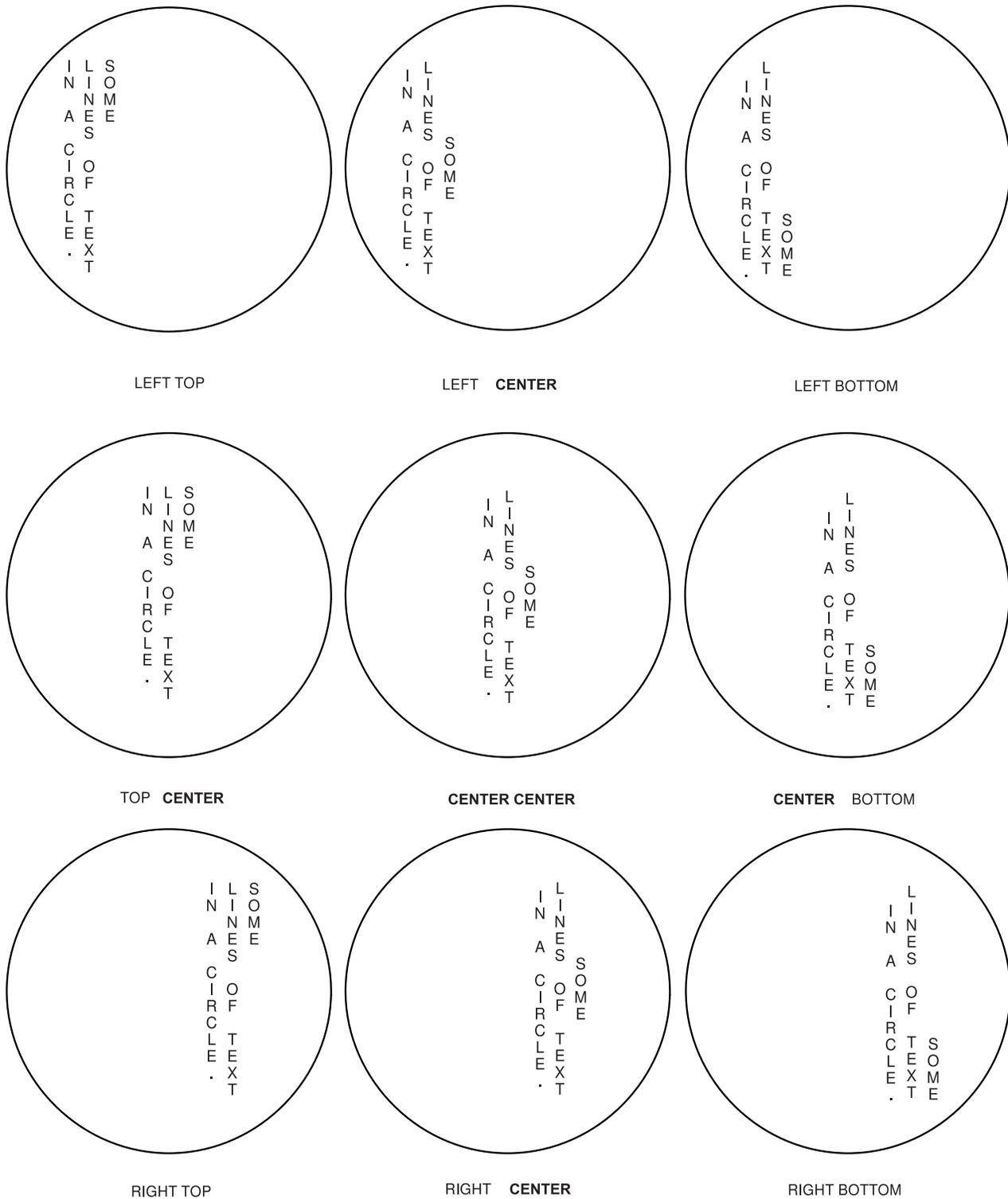


Figure 79. Placement of Text Written in **TATE** Format in a **SQUARE** Text Margin. This example is approximate.

Text Orientation

After you select the placement option, select the orientation of the text relative to the overlay. You can choose 0° (default), **90**°, **180**°, or **270**°. Do not enter the degree symbol (°).

What is the orientation of the text in Figure 95 on page 100? If you said **0**° and **270**°, you are right. The word “NOTICE” is in the 0° orientation, and the rest of the text is in the **270**° orientation. Each **WITHTEXT** subcommand allows only one orientation. You can write more than one **WITHTEXT** subcommand in a single **DRAWBOX** or **DRAWCIRCLE** command. Therefore, the **DRAWBOX** command for Box 4 requires two **WITHTEXT** subcommands.

If you choose an orientation other than 0°, the entire text block is rotated. The positioning options (**LEFT**, **RIGHT**, **TOP**, **BOTTOM**, and **CENTER**) operate relative to the text block in its new orientation. Figure 81 on page 86 shows a block of **MODERN** text at the **TOPLEFT** position in all four orientations. Notice how the meaning of **TOPLEFT** changes with the orientation. For example, the text block orientated at **90**° is positioned at the top-right corner of the box in relation to the orientation of the overlay.

Text positioning works differently for **ROUNDED** and **SQUARE** margins. For example, if you select **TOP CENTER** within a **ROUNDED** margin, the two top corners of the text block touch the **ROUNDED** margin. The text block is positioned as high as possible within the **ROUNDED** margin. All **CENTER** placements, except **CENTER CENTER**, work in a similar way.

The following placement options have the same results in a **SQUARE** or **ROUNDED** margin:

TOPLEFT
TOP RIGHT
CENTER CENTER
BOTTOM LEFT
BOTTOM RIGHT

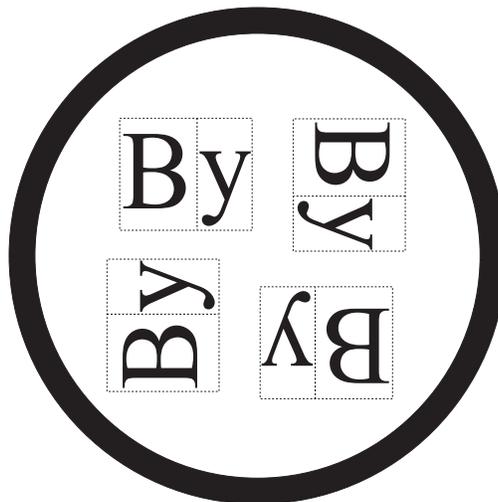


Figure 80. Text Block Orientations for a Circle

After a text block is placed, select the orientation relative to the overlay. Choose from:

0 (Default)
90
180
270

Note: The entire text block is rotated.

Positioning and Spacing the Text

The positioning options operate relative to the text block in its new orientation. Figure 80 on page 85 and Figure 81 show a block of **MODERN** text at the top-left position, in all four orientations.

Note: The position of **TOPLEFT** text changes with the orientation. For example, the text block orientated at 90° is positioned at the top-right of the circle in relation to the orientation of the overlay.

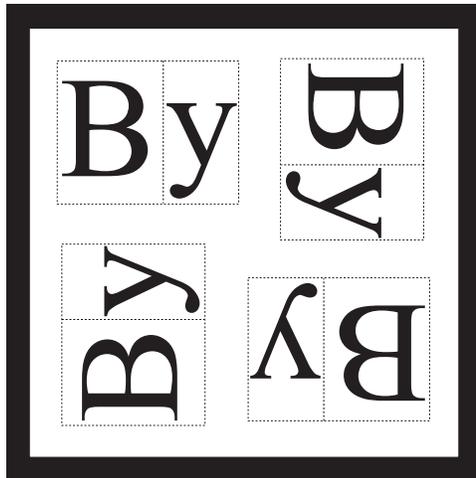


Figure 81. Text Block Orientations for a Box

Text Placement

When defining text, you specify the orientation, format, and placement of the text. There are two additional choices when specifying text placement: **BALANCE** and **JUSTIFY**.

placement Define where the text is placed in the box:

MODERN After specifying **MODERN** format, choose one placement area from the following list:

TOP
CENTER (Default)
BOTTOM

Then, choose one placement area from this list:

TOP
CENTER (Default)
BOTTOM

Note: It is the character box that is placed against the boundary of the text block. Characters in most fonts have space around the character inside the character box, so there could be some space between the character and the boundary of the text block, as indicated in Figure 81.

BALANCE Spaces the individual characters equally between the left and the right sides of the text block.

To set the width of the text block, OGL/370 draws an invisible line across the top of the block and another across the bottom. Each line is extended out until it meets the inside of the box border. If these two lines are of equal length, OGL/370 sets the width of the text block to that length. Otherwise, OGL/370 sets the block width to the length of the shorter line.⁶

6. To include blanks in balanced text, see "Blanks in Balanced Text" on page 96.

JUSTIFY

Text is left and right justified by increasing the spaces between the words as necessary to fill the text block. Leading and trailing blanks are not changed.

OGL/370 makes the text block just wide enough for both ends to touch the **SQUARE** text margin.

If you choose **JUSTIFY**, you have another placement option:

LASTNO

If you specify **JUSTIFY LASTNO**, the last text string in the text block is **LEFT** justified, but not **RIGHT** justified.

In Figure 82 and Figure 83 on page 88 **JUSTIFY** is used with the **LASTNO** option.

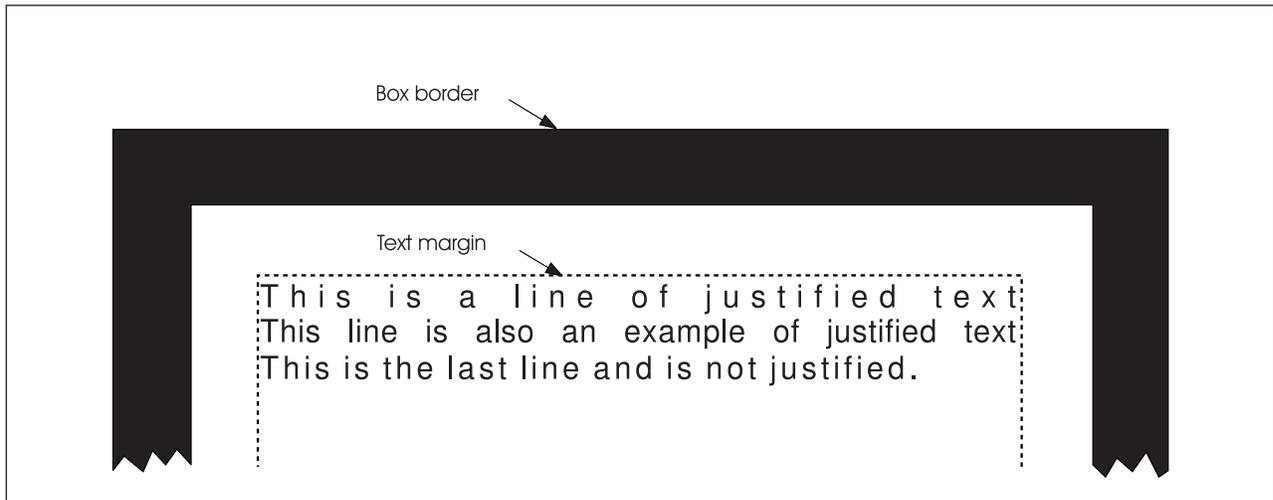


Figure 82. Justified Text in a Box

Positioning and Spacing the Text

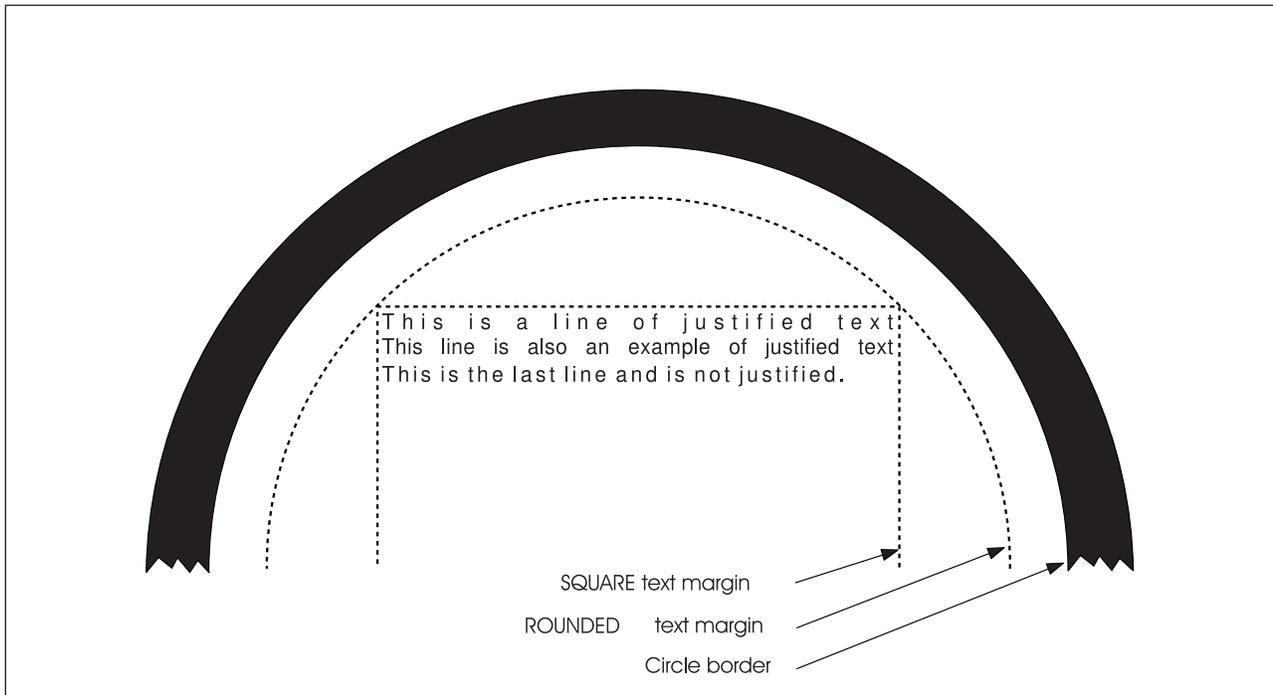


Figure 83. Justified Text in a Circle

Figure 84 on page 89 and Figure 85 on page 90 illustrates each of the balanced and justified text combinations for the **MODERN** format.

Positioning and Spacing the Text

SOME LINES OF
 TEXT A PRINTED
 IN BOX.

TOP JUSTIFY

S O M E
 P R I N T E D
 L I N E S .

TOP BALANCE

SOME LINES OF
 TEXT A PRINTED
 IN BOX.

CENTER JUSTIFY

S O M E
 P R I N T E D
 L I N E S .

CENTER BALANCE

SOME LINES OF
 TEXT A PRINTED
 IN BOX.

BOTTOM JUSTIFY

S O M E
 P R I N T E D
 L I N E S .

BOTTOM BALANCE

Figure 84. Placement of Balanced and Justified Text, Written in **MODERN** Format. This example is approximate.

Positioning and Spacing the Text

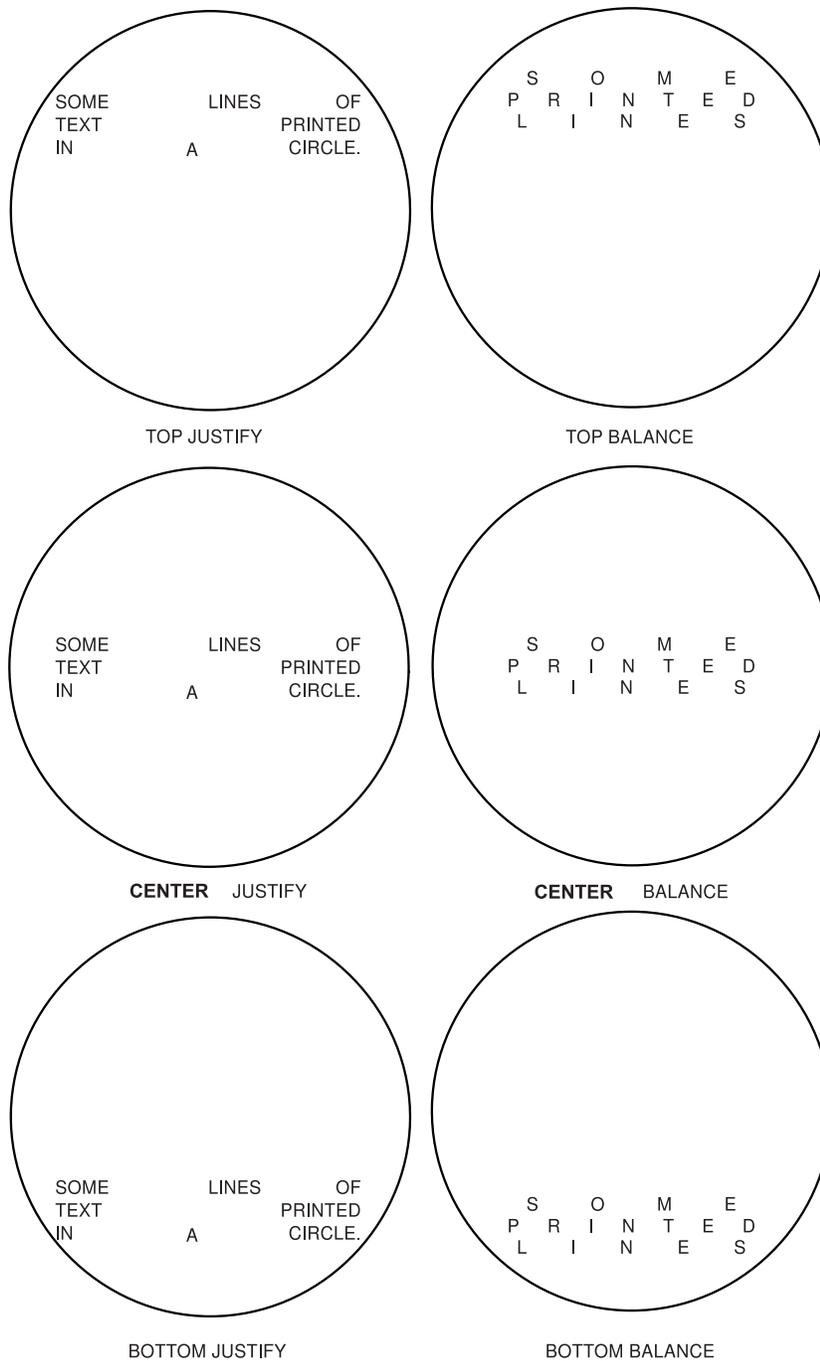


Figure 85. Placement of Balanced and Justified Text Written in **MODERN** Format in a **SQUARE** Text Margin. This example is approximate.

COLUMN and TATE

After specifying either **COLUMN** or **TATE** format, choose one placement area from the following list:

LEFT

CENTER (Default)

RIGHT

Then, choose one area from this list:

TOP

CENTER	(Default)
BOTTOM	
BALANCE	Equally spaces the characters between the top and the bottom of the text block. To set the height of the text block, OGL/370 draws an invisible line along the left side of the block, and another along the right side. Each line is extended out until it meets the inside of the box border. If these two lines are of equal length, OGL/370 sets the height of the text block to that length. Otherwise, OGL/370 sets the block height to the length of the shorter line. ⁷
JUSTIFY	Text is top and bottom justified by increasing the spaces between the words as necessary to fill the text block. Leading and trailing blanks are not changed. OGL/370 makes the text block just tall enough for both ends to touch the SQUARE text margin. If you choose JUSTIFY , you have another placement option: LASTNO If you specify JUSTIFY LASTNO , the last text string in the text block is TOP justified, but not BOTTOM justified.

Note: Placement descriptions apply to both **COLUMN** and **TATE**.

Figure 86 on page 92 and Figure 88 on page 94 illustrates each of the balanced and justified text combinations for the **COLUMN** format.

Figure 87 on page 93. and Figure 89 on page 95 illustrates each of the balanced and justified text combinations for the **TATE** format.

7. To include blanks in balanced text, see “Blanks in Balanced Text” on page 96.

Positioning and Spacing the Text

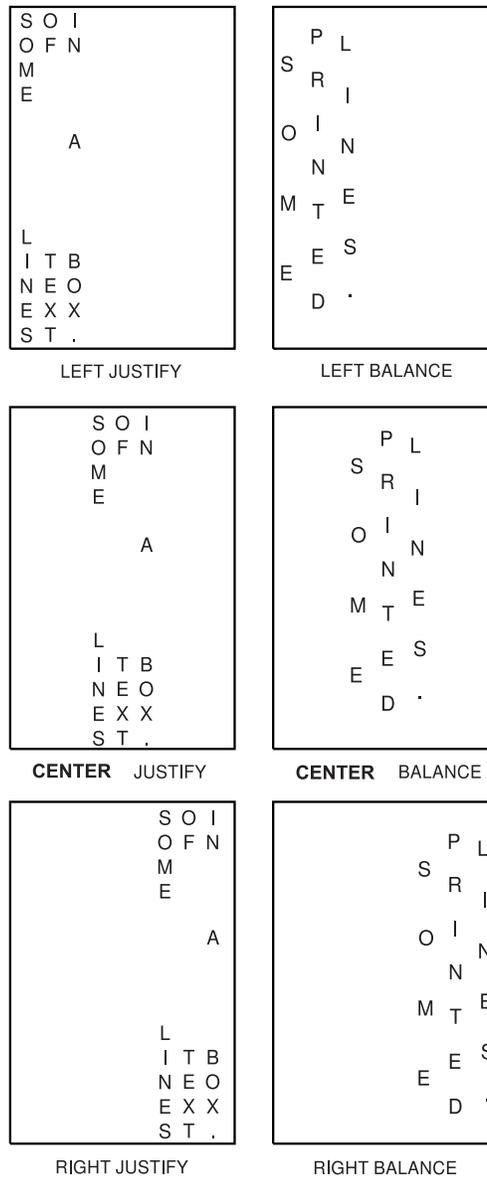


Figure 86. Placement of Balanced and Justified Text Written in **COLUMN** Format. This example is approximate.

Positioning and Spacing the Text

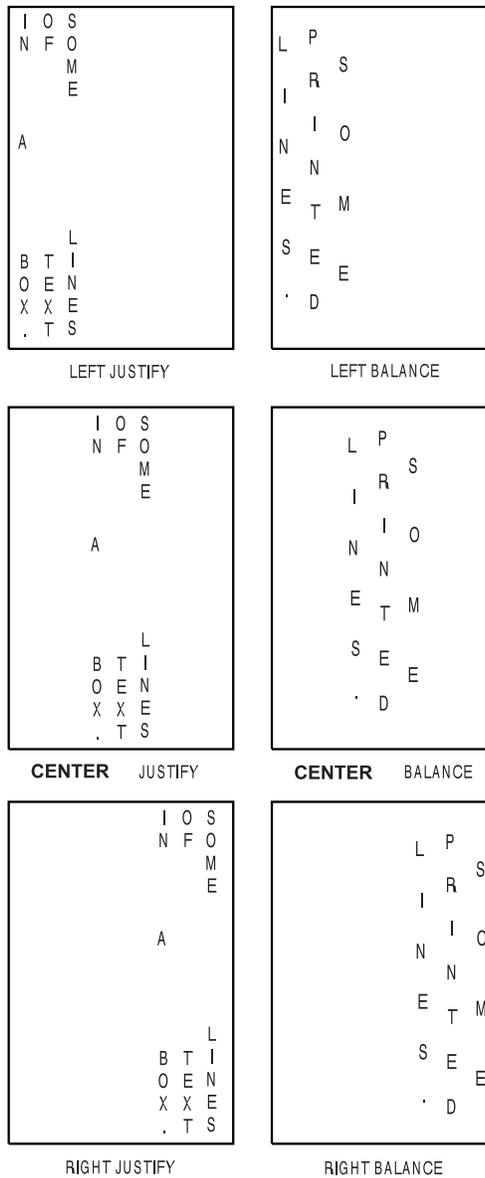
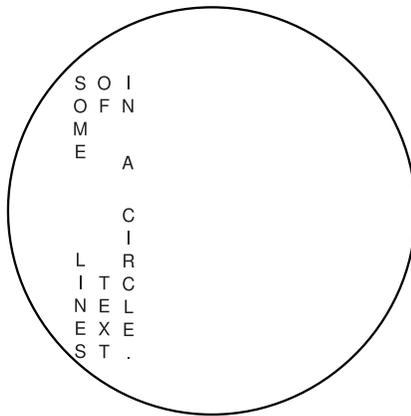
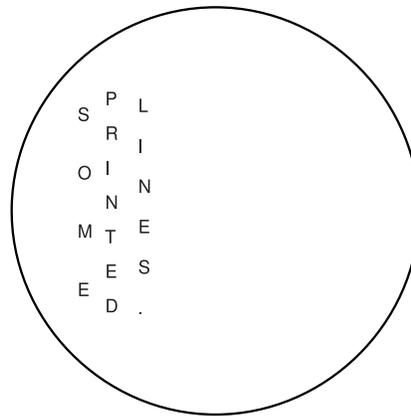


Figure 87. Placement of Balanced and Justified Text Written in **TATE** Format. This example is approximate.

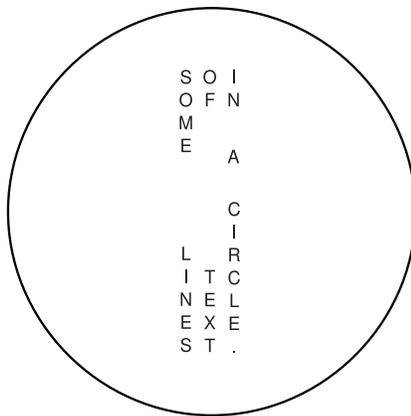
Positioning and Spacing the Text



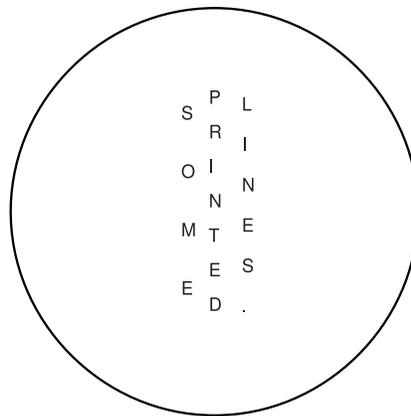
LEFT JUSTIFY



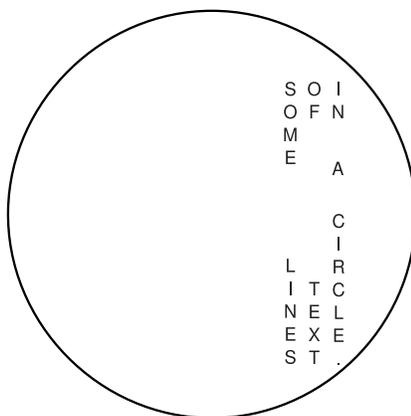
LEFT BALANCE



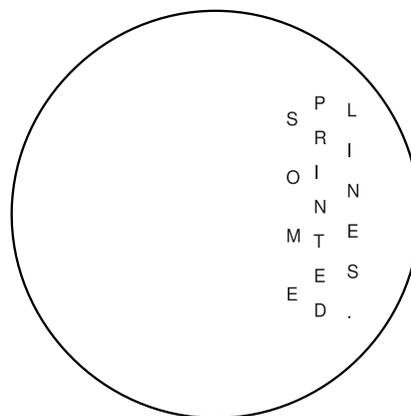
CENTER JUSTIFY



CENTER BALANCE



RIGHT JUSTIFY



RIGHT BALANCE

Figure 88. Placement of Balanced and Justified Text Written in **COLUMN** Format in a **SQUARE** Text Margin. This example is approximate.

Positioning and Spacing the Text

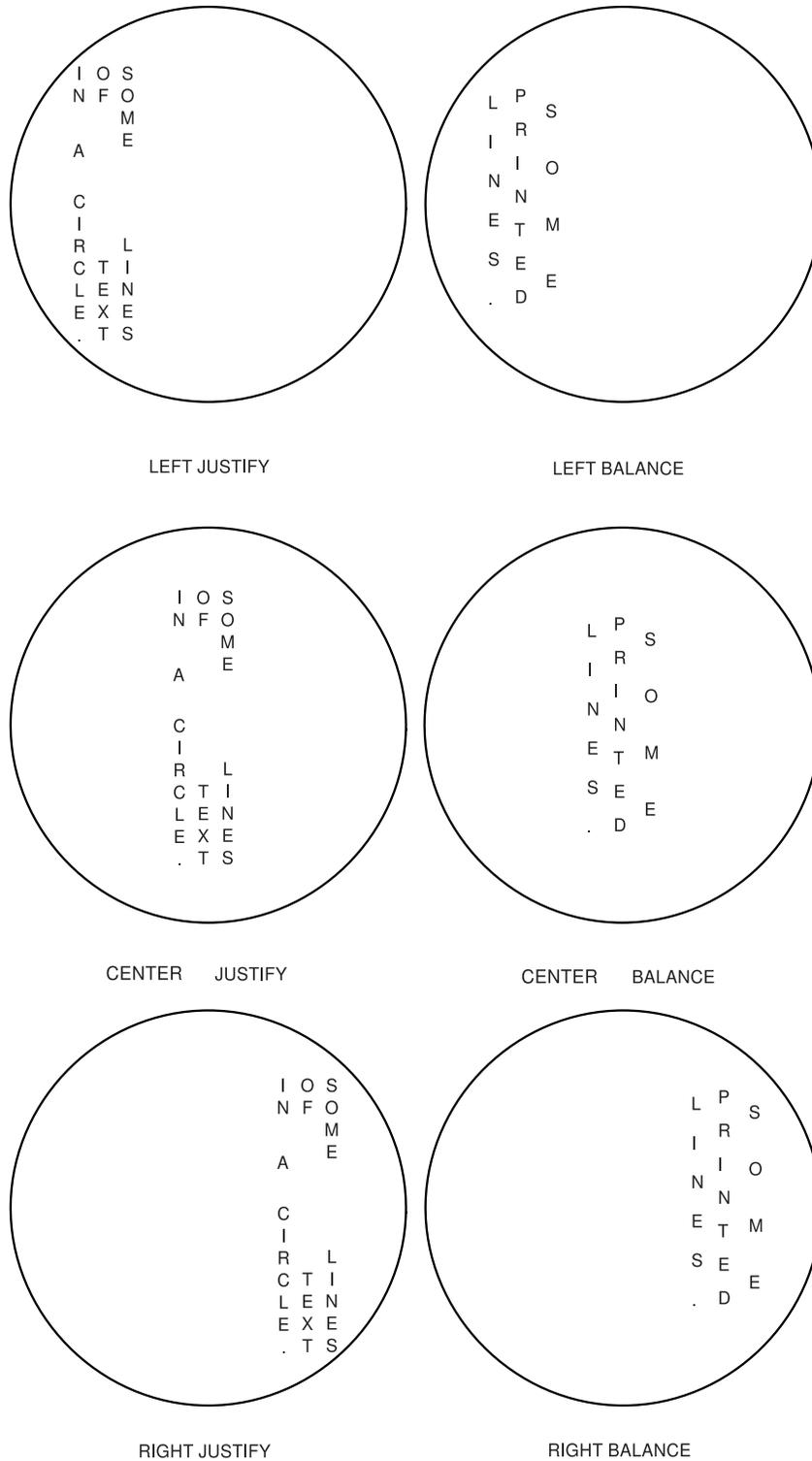


Figure 89. Placement of Balanced and Justified Text Written in **TATE** Format in a **SQUARE** Text Margin. This example is approximate.

Positioning and Spacing the Text

Blanks in Balanced Text

If you have specified **BALANCE** for the text placement, you should not include blanks between the apostrophes. However, if you want to include blanks in text that is balanced, you can do it by specifying multiple text segments: one segment for each character string (excluding blanks) and one segment for each string of blanks. Let us say that you wanted to print one of the following:



Figure 90. *Balanced Text With Blanks. This example is approximate.*

The following command produces the first box:

```
DRAWBOX 5 IN .75 IN MEDIUM SOLID
WITHTEXT 0 MODERN CENTER BALANCE
LINE FONTX NOUNDERLINE CHAR 'Some'
NOUNDERLINE CHAR ' '
NOUNDERLINE CHAR 'balanced'
NOUNDERLINE CHAR ' '
NOUNDERLINE CHAR 'text.';
```

For the second and third boxes in Figure 90, all you need to change is the box size. Notice that two of the text segments specify one blank space each. The space taken up by a blank is the same as the space between the characters in the words “Some”, “balanced”, and “text”.

You now have all the information you need to write the **WITHTEXT** subcommands for the text shown in Figure 95 on page 100. Try it, and compare your results with Figure 91 on page 97.

WITHTEXT	270	<u>MODERN</u>	TOP	<u>CENTER</u>	SPACED .2 in
subcommand word	orientation	format	placement	line spacing	
LINE	font3	<u>NOUNDERLINE</u>	<u>CHAR</u>	'Keep this copy.'	
LINE	font3	<u>NOUNDERLINE</u>	<u>CHAR</u>	'NOT VALID as ID.'	
line option	font name	underlining option	text type	text	
WITHTEXT	<u>0</u>	COLUMN	RIGHT	BALANCE	
subcommand word	orientation	format	placement		
LINE	font2	<u>NOUNDERLINE</u>	<u>CHAR</u>	'NOTICE'	;
line option	font name	underlining option	text type	text	end marker

Figure 91. WITHTEXT Command

Line spacing: Line spacing is the distance from the baseline of one text string to the baseline of the next. All of the examples presented in Figure 74 on page 79, Figure 76 on page 81, and Figure 78 on page 83. contain more than one text string. If a text block has two or more text strings, you have two choices:

AUTO

Standard line spacing for the font being used. If the text is printed in more than one font, the default line spacing for any two text strings is based on the largest font used in those strings (default).

SPACED

If you enter **SPACED**, you must specify the distance from the baseline of one text string to the baseline of the next. For text printed in **MODERN** format, the baseline is an invisible line on which the characters rest. For text printed in **COLUMN** or **TATE** formats, the baseline is an invisible line drawn through the center of a text string, see Figure 92 on page 98. If you specify the distance between strings, leave enough space so text does not look crowded.

Specify the distance with a number (*n*) and a unit of measurement:

n

- IN** Inches
- MM** Millimeters
- PELS** Pels
- POINTS** Points
- LPI** Lines per inch

Notes:

1. A point is a unit of measurement used by typesetters. There are 72 points in an inch.
2. **LPI** is an abbreviation for lines per inch. If you specify 10 **LPI** for a block of 10 text strings, the distance from the top of the first text string to the top of the 11th text string, is 1 inch.

Positioning and Spacing the Text

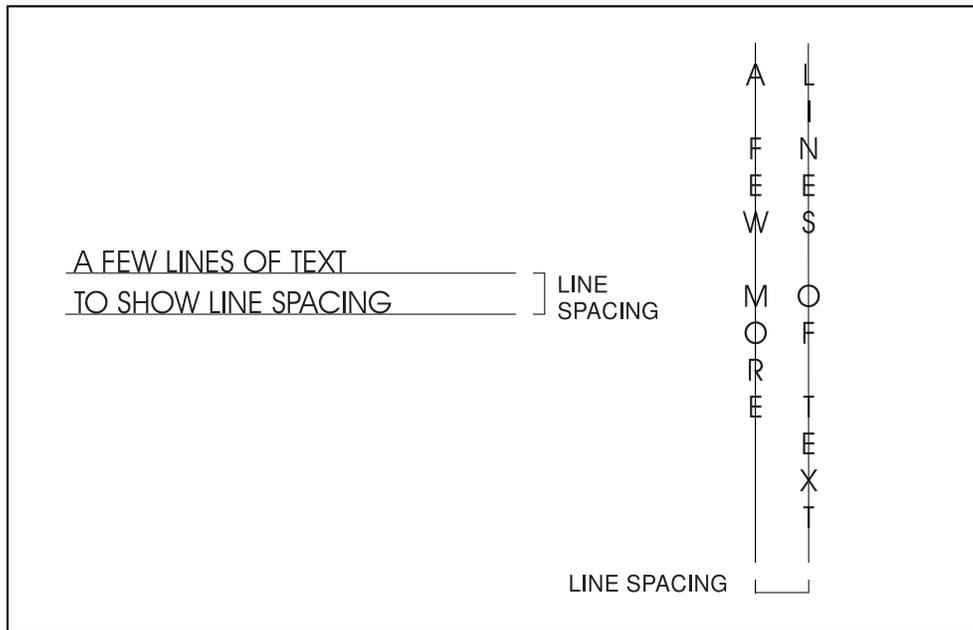


Figure 92. Line Spacing in a Box

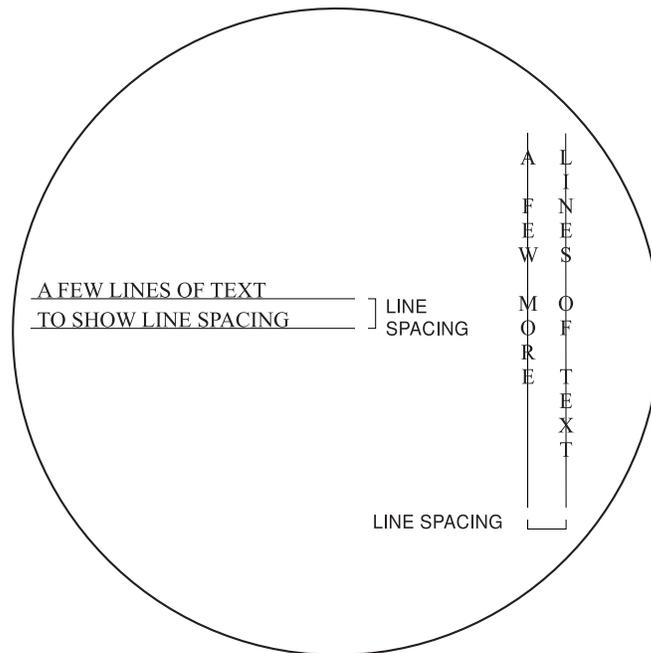


Figure 93. Line Spacing in a Circle

Ensure the Text Fits

When OGL/370 places text on an overlay, it ensures the text fits within the boundaries of the overlay as defined in the **OVERLAY** command. If any of the text string extends off the overlay, none of the text string is printed. If the text is in a box or circle (visible or invisible) and some of it does not fit within the text margin, but does fit in the overlay, it is printed. In either case, you get a message.

To understand the message, it helps to know how OGL/370 determines if the text fits. The characters in a font are seen as rectangles called *character boxes*. OGL/370 finds the physical length of the text string by adding the character box sizes for each character in the string. OGL/370 also considers the font point size. The point size is the sum of two values:

Maximum ascender: The distance from the top of the tallest character in the font to the baseline.

Maximum descender: The distance from the baseline to the bottom of the lowest character (for example, the tail of a lowercase “y”).

For text in a particular font to fit, neither the physical length of the string nor the point size of the character can exceed the boundaries of the box or the overlay.

Figure 94 illustrates how maximum ascenders and maximum descenders can exceed text boundaries.

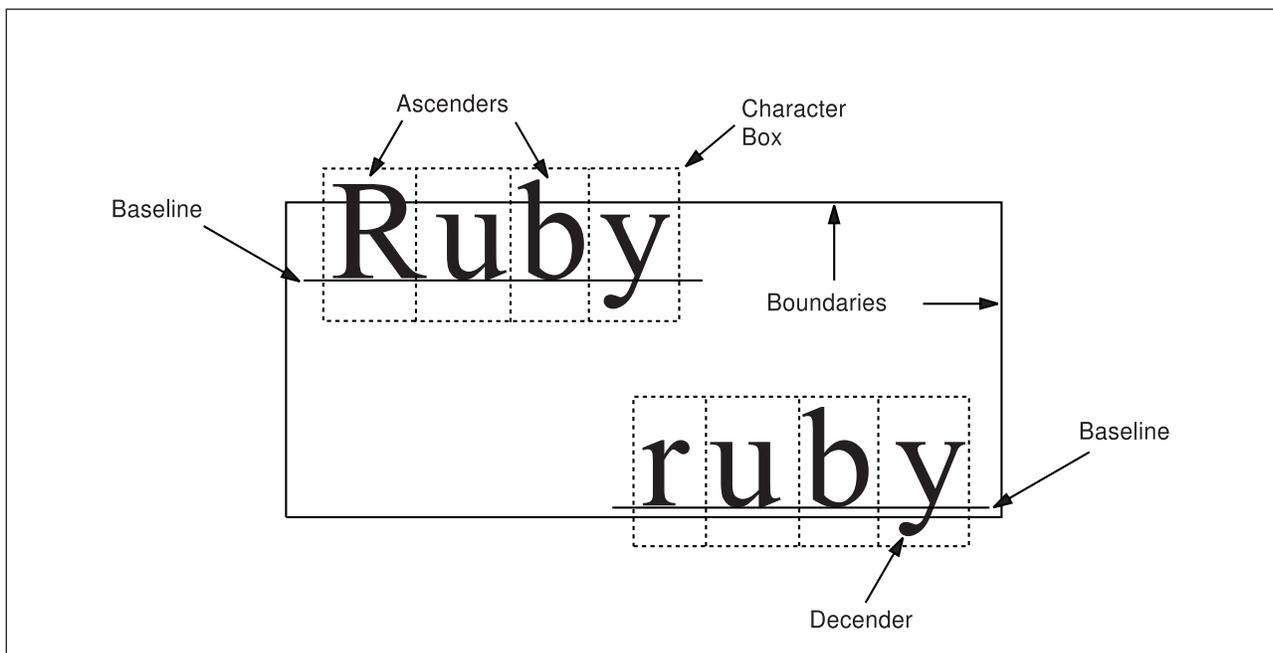


Figure 94. Text Exceeding Boundaries

Right or left kerning can also be the cause for text extending beyond the overlay boundary. In this case, the text does not print and no message is generated by OGL/370. However, you might get a PSF diagnostic message.

Adding Text to Boxes

Adding Text to Boxes (DRAWBOX WITHTEXT)

In overlay “RCPT”, shown in Figure 14 on page 23, the box with the dashed border has text in it. Use the **DRAWBOX** command and the **WITHTEXT** subcommand to draw a box and include text. First you must name the fonts you want to use in your overlay, by using the **FONT** command.

In the overlay sample, the box mentioned above looks like this:

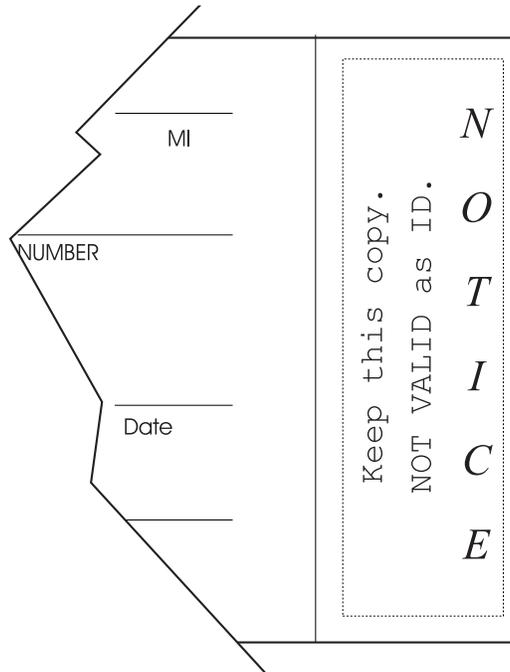


Figure 95. Box with Text

The **DRAWBOX** command for the box shown in Figure 95 is:

```
DRAWBOX .7 IN 2.4 IN BOLD DASHED;
```

To print words, you must include the **WITHTEXT** subcommand, which specifies:

- Orientation, format, and placement of text
- Line spacing
- Fonts to be used
- Text to be printed.

Using the **WITHTEXT** subcommand, position and space the 270° text in Figure 95. Space the lines 0.2 inches apart, (**SPACED .2 IN**). Then compare your version with Figure 96 on page 101.

The following is an example of a **WITHTEXT** subcommand used for some of the text in Figure 95:

WITHTEXT	270	<u>MODERN</u>	TOP <u>CENTER</u>	SPACED .2 in
subcommand word	orientation	format	placement	line spacing
LINE	font3	<u>NOUNDERLINE</u>	<u>CHAR</u>	'Keep this copy.'
LINE	font3	<u>NOUNDERLINE</u>	<u>CHAR</u>	'NOT VALID as ID.';
subcmd word	font name	underlining option	text type	text

Figure 96. WITHTEXT Subcommand

subcommand word

WITHTEXT

orientation The text is oriented relative to the overlay. Choose from:

- 0**
- 90**
- 180**
- 270**

format The format in which the text is written. See “Positioning and Spacing the Text” on page 76 for descriptions of the formats. Choose from:

- MODERN (Default)
- COLUMN**
- TATE**

Adding Text without Boxes

Most of the text in overlay “RCPT” (Figure 14 on page 23) is not in boxes. In fact, there is only one instance of text written in a box.

To write text without boxes around it, use one of two commands:

SETTEXT

Described in “Adding Text Another Way (**SETTEXT**)” on page 108.

DRAWBOX

Described in this section.

Consider what happens if in the **DRAWBOX** command for the last example, you were to write the first line like this:

```
DRAWBOX .7 IN 2.4 IN 0 DASHED
```

From the prior **DRAWBOX** example in “Adding Text to Boxes (**DRAWBOX WITHTEXT**)” on page 100, the only change is that the border thickness is 0 pels rather than **BOLD** (6 pels). In other words, the border is invisible. The result of this new command is illustrated in Figure 97 on page 102.

Because an invisible dashed border looks just like an invisible dotted or solid border, you do not have to define the border type. You can rewrite the first line like this and get the same result:

Adding Text to Boxes

DRAWBOX .7 IN 2.4 IN 0

N
O
T
I
C
E

Keep this copy.
NOT VALID as ID.

Figure 97. An Invisible Box with Text

For text aligned along borders, the distance between the text and the border is equal to the thickness of the border. For a border thickness of 0, the text begins on the border. With this new way of seeing borders (or not seeing them), look at Figure 14 on page 23 and find the “invisible boxes” around the text. Because the boxes are invisible, different people may specify different size boxes. For example, if we represent invisible boxes with dotted rules, the top of overlay “RCPT” may be seen as the illustration shown in Figure 98:

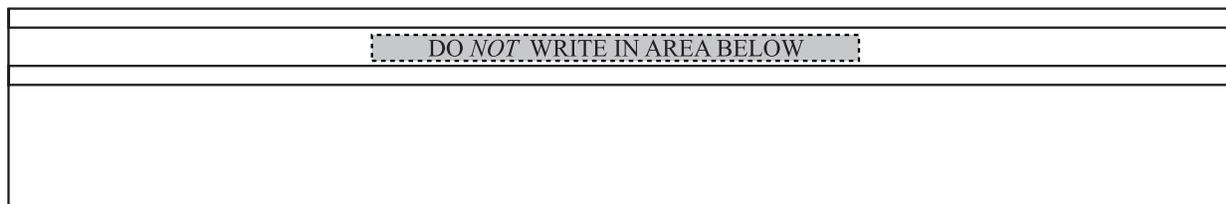


Figure 98. Text Placement Within an Invisible Box Version 1

Or it may be seen as the illustration shown in Figure 99:

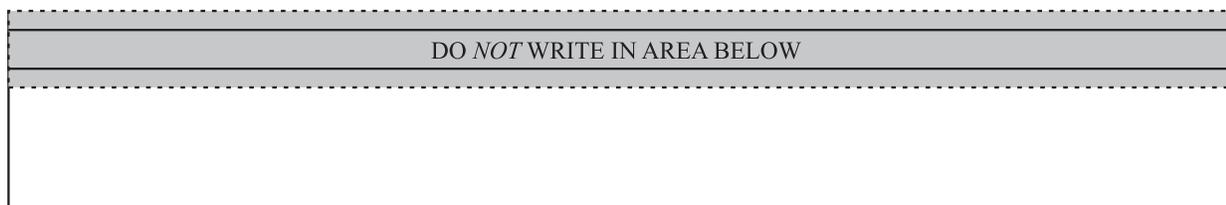


Figure 99. Text Placement Within an Invisible Box Version 2

Either invisible box, if correctly placed, reproduces the top of the overlay. However, placing the first box is more difficult.

To place the box in version 1, you would need to know the printed width and length of the text string before you could even begin to place the invisible box.

On the other hand, for the box in version two, you define the box as wide as the overlay and as high as the distance between the top horizontal rule and the second horizontal rule below the text. Specify the

Adding Text to Boxes

shading of the box. Then position the text at **MODERN CENTER CENTER**, and OGL/370 does all the figuring for you. The commands for the second box are:

```
POSITION ABSOLUTE 0 ABSOLUTE 0;  
DRAWBOX 7 IN .4 IN 0 SHADE LIGHT  
WITHTEXT 0 MODERN CENTER CENTER  
  LINE FONT1 NOUNDERLINE CHAR 'DO '  
  FONT2 NOUNDERLINE CHAR 'NOT '  
  FONT1 NOUNDERLINE CHAR 'WRITE IN AREA BELOW';
```

Note: You can shade the signature area in overlay “RCPT” by using an invisible box.

Referring to the invisible boxes illustrated in Figure 100, write the commands to position and define the text. Include the commands for the shaded signature area. Use standard line spacing for the name and title of the college president. Compare your results with the example in “**DRAWBOX WITHTEXT** Commands for Overlay RCPT” on page 104.

DO NOT WRITE IN AREA BELOW

Total
Units

Registration
Out-of-State
Other
Total

***TropiCal
Community College***

LAST NAME FIRST NAME MI

SOCIAL SECURITY NUMBER

Received by

Date

Robinson K. Russo
President

Keep this copy.
NOT VALID as ID.

Figure 100. Printed Text. The invisible boxes around the text are represented by the dotted rules.

Adding Text to Boxes

DRAWBOX WITHTEXT Commands for Overlay RCPT

```
-'ADDING WORDS'  
POSITION ABSOLUTE 6.2 IN ABSOLUTE .5 IN;  
DRAWBOX .7 IN 2.4 IN BOLD DASHED  
  WITHTEXT 270 MODERN TOP CENTER SPACED .2 IN  
    LINE FONT3 NOUNDERLINE CHAR 'Keep this copy.'  
    LINE FONT3 NOUNDERLINE CHAR 'NOT VALID as ID.';  
  
POSITION ABSOLUTE 0 ABSOLUTE 0;                               -HEADER  
DRAWBOX 7 IN .4 IN 0 SHADE STANDARD LIGHT  
  WITHTEXT 0 MODERN CENTER CENTER  
    LINE FONT1 NOUNDERLINE CHAR 'DO '  
    FONT2 NOUNDERLINE CHAR 'NOT '  
    FONT1 NOUNDERLINE CHAR 'WRITE IN AREA BELOW';  
  
POSITION ABSOLUTE .25 IN ABSOLUTE 1.1 IN;                   -UNITS  
DRAWBOX .5 IN .5 IN 0  
  WITHTEXT 0 MODERN TOPLEFT SPACED .15 IN  
    LINE FONT5 NOUNDERLINE CHAR 'Total'  
    LINE FONT5 NOUNDERLINE CHAR 'Units';  
  
POSITION ABSOLUTE 1.9 IN ABSOLUTE .5 IN;                     -FEES  
DRAWBOX 1 IN 1 IN 0  
  WITHTEXT 0 MODERN BOTTOM LEFT SPACED .25 IN  
    LINE FONT5 NOUNDERLINE CHAR 'Registration'  
    LINE FONT5 NOUNDERLINE CHAR 'Out-of-State'  
    LINE FONT5 NOUNDERLINE CHAR 'Other'  
    LINE FONT5 UNDERLINE CHAR 'Total';  
  
POSITION ABSOLUTE 3.25 IN ABSOLUTE .75 IN;                   -NAME  
DRAWBOX 2.5 IN .2 IN 0  
  WITHTEXT 0 MODERN BOTTOM LEFT  
    LINE FONT5 NOUNDERLINE CHAR  
      'LAST NAME      FIRST NAME      MI';  
  
POSITION ABSOLUTE 3.25 IN ABSOLUTE 1.25 IN;                   -'SOC. SEC. NO.'  
DRAWBOX 2.5 IN .2 IN 0  
  WITHTEXT 0 MODERN BOTTOM CENTER  
    LINE FONT5 NOUNDERLINE CHAR 'SOCIAL SECURITY NUMBER';  
  
POSITION ABSOLUTE 3.25 IN ABSOLUTE 2 IN;                       -RECEIVED  
DRAWBOX 1 IN .25 IN 0  
  WITHTEXT 0 MODERN LEFT CENTER  
    LINE FONT5 NOUNDERLINE CHAR 'Received by';  
POSITION ABSOLUTE 5.25 IN ABSOLUTE 2 IN;                       -DATE  
DRAWBOX .5 IN .25 IN 0  
  WITHTEXT 0 MODERN LEFT CENTER  
    LINE FONT5 NOUNDERLINE CHAR 'Date';  
POSITION ABSOLUTE 1.25 IN ABSOLUTE 1.75 IN;                   -COLLEGE  
DRAWBOX 1.75 IN .6 IN 0  
  WITHTEXT 0 MODERN CENTER CENTER SPACED .2 IN  
    LINE FONT2 NOUNDERLINE CHAR 'TropiCal'  
    LINE FONT2 NOUNDERLINE CHAR 'Community College';  
  
POSITION ABSOLUTE 3.25 IN ABSOLUTE 2.65 IN;                   -PRESIDENT  
DRAWBOX 1.25 IN .5 IN 0  
  WITHTEXT 0 MODERN LEFT TOP SPACED .2 IN  
    LINE FONT5 NOUNDERLINE CHAR 'Robinson K. Russo'  
    LINE FONT5 NOUNDERLINE CHAR 'President';
```

Adding Text to Boxes

You are now ready to send your definition to be processed by OGL/370. Your printout should look like the illustration in Figure 101.

DO <i>NOT</i> WRITE IN AREA BELOW		
<input type="checkbox"/>		
Total		Registration
Units		Out-of-State
		Other
		<u>Total</u>
<i>TropiCal</i> <i>Community College</i>		
	LAST NAME	FIRST NAME MI
	-	-
	SOCIAL SECURITY NUMBER	
	Received by	Date
	Robinson K. Russo	
	President	
		Keep this copy. NOT VALID as ID.

Figure 101. Sample Overlay without Graphics

The following is the complete **DRAWBOX** command for the box in Figure 95 on page 100:

```
DRAWBOX .7 IN 2.4 IN BOLD DASHED
  WITHTEXT 270 MODERN TOP CENTER SPACED .2 IN
    LINE FONT3 NOUNDERLINE CHAR 'Keep this copy.'
    LINE FONT3 NOUNDERLINE CHAR 'NOT VALID as ID.'
  WITHTEXT 0 COLUMN RIGHT BALANCE
    LINE FONT2 NOUNDERLINE CHAR 'NOTICE';
```

Note: The end marker (;) appears only at the end of the entire command.

Adding Text to Circles

Adding Text to Circles (DRAWCIRCLE WITHTEXT)

To print text in a circle, use the **WITHTEXT** subcommand.

The **WITHTEXT** subcommand for **DRAWCIRCLE** is the same as **DRAWBOX**. Figure 102 illustrates a circle with text.

An example of a **DRAWCIRCLE** command is:

```
DRAWCIRCLE 1.0 IN MEDIUM SOLID;
```

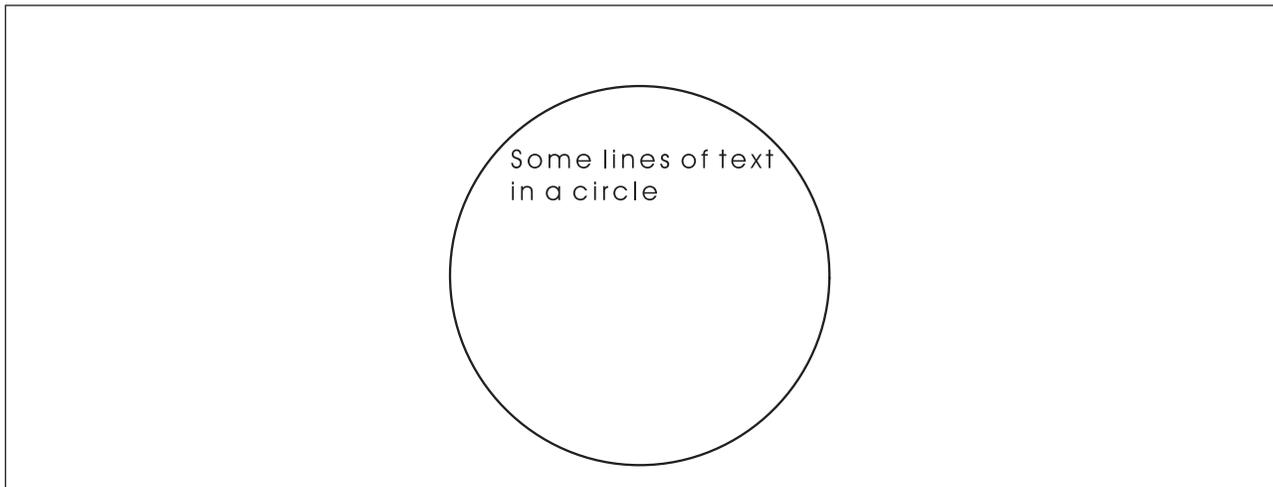


Figure 102. Circle with Text

Specifying the Text for DRAWBOX and DRAWCIRCLE

Next you describe the text.

subcommand word

To begin defining a *text string*, you must enter the word:

LINE

This subcommand can be used more than once in a **WITHTEXT** subcommand. Each use marks the beginning of a new text string.

font name

This is described further in Chapter 4, "Adding Text" on page 65.

underlining

Indicate whether you want the following text segment underlined. Blank spaces included as part of the text, are also underlined. Choose from:

NOUNDERLINE

Does not underline the following text segment (default).

UNDERLINE

Underline the following text segment.

Notes:

1. This option is only effective for the text segment it precedes.
2. Only text written in **MODERN** format can be underlined.

text type

Choose from:

CHAR *Character text:* The characters you type are the characters that are printed on your overlay. Most text is entered as character text (default).

HEX *Hexadecimal text:* If you want to include characters that do not appear on your keyboard, you enter the code points for those characters. This method is explained in the *text type* section of “Adding Text Another Way (SETTEXT)” on page 108.

text Enter the text, enclosed in apostrophes. For example:

'Keep this copy'

Note: If the text contains an apostrophe or a semicolon (;), represent it with two apostrophes or semicolons, respectively. For example, to print the text:
enter:

'Walk; don''t run'

'Walk;; don''t run'

Notes:

1. If a text string extends beyond the overlay, it is not printed.
2. If a text string extends beyond the box, but not beyond the overlay, it is printed.

end marker Always end a command with an end marker (;).

Adding Text Another Way

Adding Text Another Way (SETTEXT)

This section describes how to include text in an overlay without placing it in a box (visible or invisible) or circle.

The syntax of the **SETTEXT** command is the same as that of the **WITHTEXT** subcommand except for the command word and the keywords that describe the text placement. The major difference is in the way text is positioned. For **SETTEXT**, you determine the origin of the text and place the text relative to that origin. The origin of the text is the intersection of two rules, see Figure 103.

1. The baseline of the first line. For text printed in **MODERN** format, the baseline is an invisible line on which the text rests. For text printed in **COLUMN** or **TATE** format, the baseline is an invisible line drawn through the center of a text string.
2. A line perpendicular to the beginning of the longest text string.

The text origins for all three formats are shown in Figure 105 on page 110, Figure 106 on page 110., and Figure 107 on page 111.

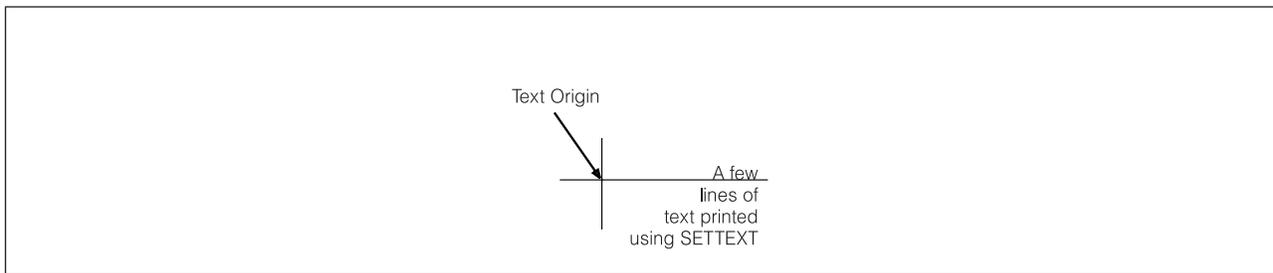


Figure 103. Text Origin with **SETTEXT**. The text origin is at the point where the vertical and horizontal rules intersect.

The **SETTEXT** command has the following entries:

SETTEXT		<u>0</u>	<u>MODERN</u>	RIGHT	SPACED .8
command		orientation	format	alignment	line spacing
LINE	font4	<u>NOUNDERLINE</u>	<u>CHAR</u>	'A few '	
LINE	font4	<u>NOUNDERLINE</u>	<u>CHAR</u>	'lines of '	
LINE	font4	<u>NOUNDERLINE</u>	<u>CHAR</u>	'text printed '	
LINE	font4	<u>NOUNDERLINE</u>	<u>CHAR</u>	'using SETTEXT'	;
line option	font name	underlining	text type	text	end marker

Figure 104. SETTEXT

command word

SETTEXT

orientation The text is oriented relative to the overlay. Choose from:

- 0** (Default)
- 90**
- 180**
- 270**

Orientation is determined exactly as it is for the **WITHTEXT** subcommand. Each **SETTEXT** command allows only one orientation.

format The format in which the text is written. You can choose one of three formats, which are the same as those for **WITHTEXT**:

- MODERN** (Default)
- COLUMN**
- TATE**

Restrictions: There are twelve possible combinations of orientation and format. Two factors might limit the combinations you use for your overlay: fonts and printers.

Your printer might not be able to print every combination. See Appendix H, "Printer Characteristics" on page 335 for details.

Your fonts might not be available for use in all combinations; ask your system programmer in what combinations your fonts are available.

alignment There are three types of text alignment that are used with **MODERN** and three that are used with **COLUMN** and **TATE**.

After **MODERN**, choose one alignment from the following list:

- LEFT** Each text string begins directly under the beginning of the first text string (default).
- RIGHT** Each text string ends directly under the end of the first text string.
- CENTER** The center of each text string is directly under the center of the first text string.

Figure 105 on page 110 presents an example of each of the three alignments for **MODERN** and their origins.

Adding Text Another Way

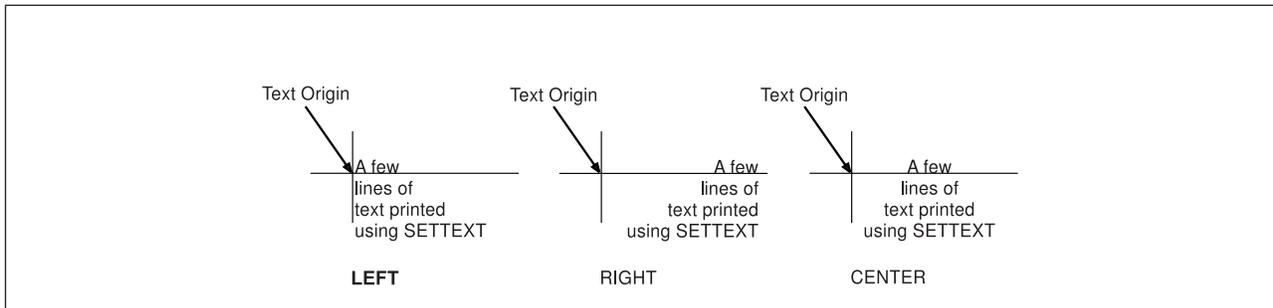


Figure 105. Alignment of Text Written with **MODERN**

After **COLUMN** or **TATE**, choose one alignment from the following list:

- TOP** The beginning of each text string is directly to the right (**COLUMN**) or left (**TATE**) of the beginning of the first text string (default).
- BOTTOM** The end of each text string is directly to the right (**COLUMN**) or left (**TATE**) of the end of the first text string.
- CENTER** The center of each text string is directly to the right (**COLUMN**) or left (**TATE**) of the center of the first text string.

Figure 106 illustrates each of the three alignments for **COLUMN** and their origins.

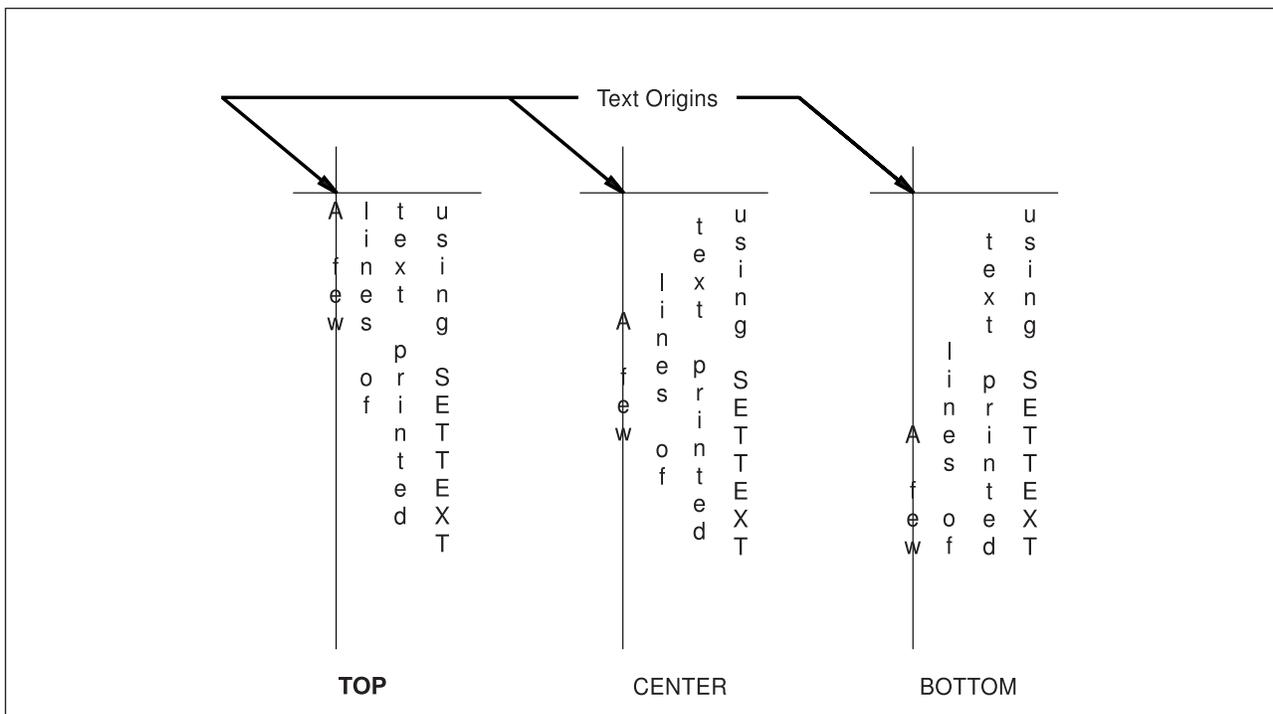


Figure 106. Alignment of Text Written with **COLUMN**

Figure 107 on page 111 illustrates each of the three alignments for **TATE** and their origins.

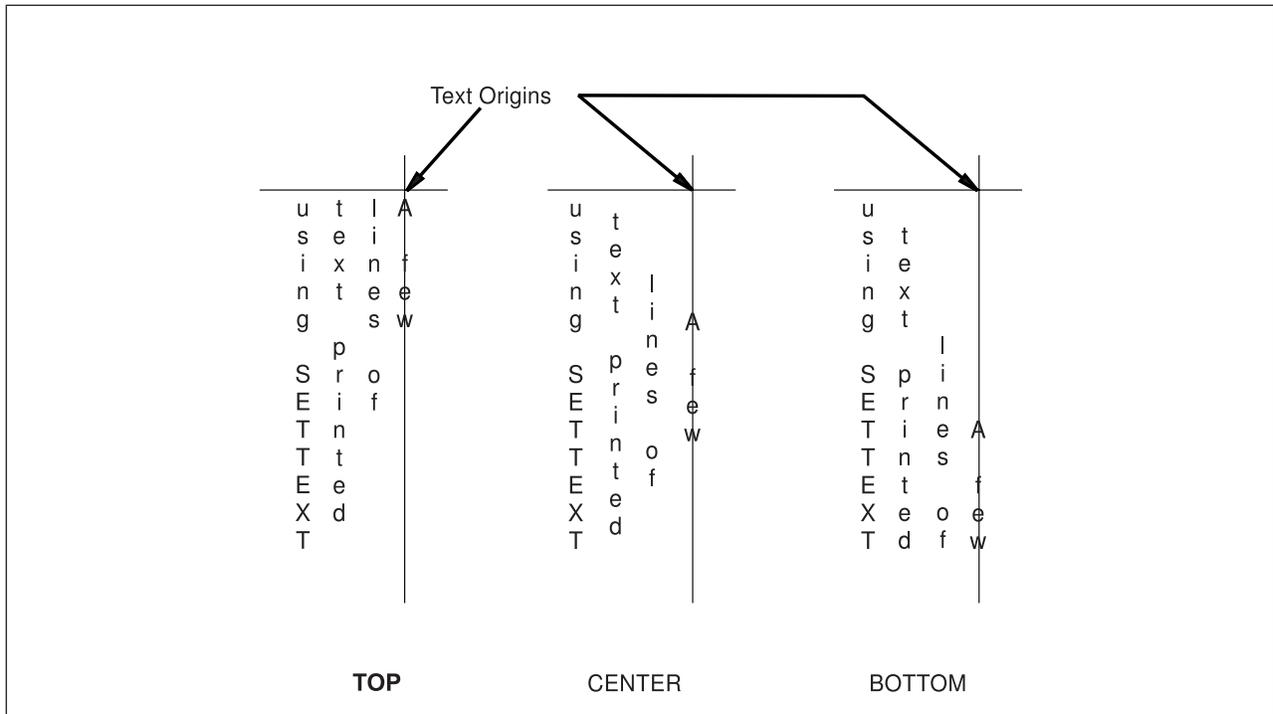


Figure 107. Alignment of Text Written with **COLUMN**

line spacing If a text block has two or more text strings, you have two choices:

AUTO The line spacing is standard line spacing for the font being used. If the text is printed in more than one font, the line spacing is based on the largest font (default).

SPACED If you enter this word, you must then specify the distance from one text string to the next (see Figure 92 on page 98).

Specify the distance with a number (*n*) and a unit of measurement. Choose from:

- n*
- IN** Inches
- MM** Millimeters
- PELS** Pels
- LPI** Lines per inch
- POINTS** Points

Or you can default to the line spacing established in **SETUNITS** (if the **LINESP** option is used).

subcommand word

To begin defining a text string, enter the word:

LINE

This subcommand can be used more than once in a **SETTEXT** command. Each use marks the beginning of a new text string.

font names This is described further in Chapter 4, “Adding Text” on page 65.

SOSI mode This option defines the way SOSI characters are to be handled. This option is ignored if **NOSOSI** is specified in the **CONTROL** command.

Adding Text Another Way

SOSI1 A single-byte character space appears wherever SOSI delimiters occur.

SOSI2 No space appears in the positions held by SOSI delimiters (default).

underlining

Indicates whether you want the following text segment underlined. Blank spaces included as part of the text segment are also underlined. Choose from:

NOUNDERLINE

Does not underline the following text segment (default).

UNDERLINE Underlines the following text segment.

Notes:

1. This option is only effective for the segment that it precedes.
2. Only text written in **MODERN** format can be underlined.

text type

Choose from:

CHAR *Character text:* The characters entered, are the characters that are printed on the overlay (default).

HEX *Hexadecimal text:* To include characters that do not appear on the keyboard, but are available with the present font, you enter the code points for those characters. The computer interprets the code and the printer prints the corresponding character.

For example, an overlay may include the following text string:

Resistance measured in ohms (Ω)

Here is one way to specify this text string:

```
SETTEXT LINE N2108C 'Resistance measured in ohms ('
              S198  HEX  '6F'
              N2108C CHAR  ')';
```

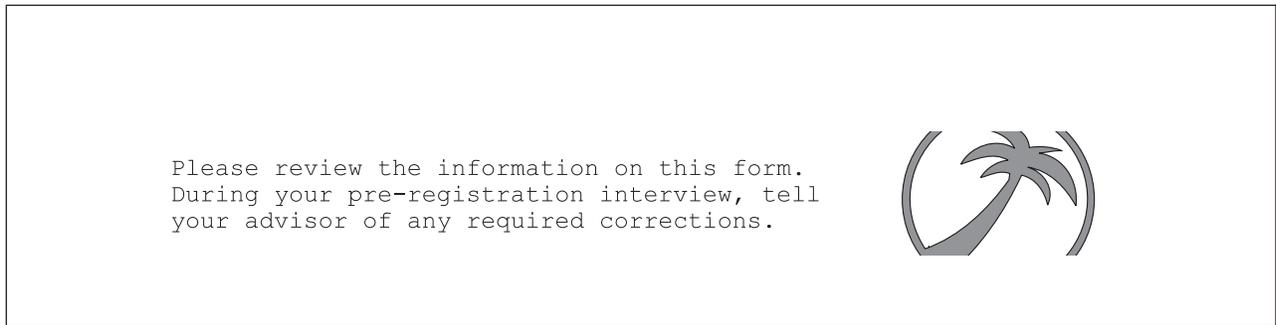
In the preceding example, the character “ Ω ” is not in font N2108C, but is in font S198. Because it is not in font N2108C, you must name the new font before entering **HEX X'6F'**.

To find the characters available in each font and their hexadecimal codes, see your system programmer.

text

An important feature that applies to **SETTEXT** and **WITHTEXT** is the use of symbolic data sets as a convenient way to include special characters in the text and to include text that may be repeated in the forms your organization uses. See “Symbolic Data Sets or Files” on page 114 for an explanation of this feature.

Because the text in Figure 108 on page 113 is written across and is **LEFT** justified, it is easily printed using **SETTEXT**.



*Figure 108. Text Printed Using **SETTEXT***

The full **SETTEXT** command for the text printed in Figure 108 is in Figure 104 on page 109. The default line spacing is .15 IN, and the **SETTEXT** command is preceded by the following **POSITION** command:

```
POSITION ABSOLUTE 1 ABSOLUTE 2;
```

Symbolic Data Sets or Files

A symbolic data set or file contains strings of text, each string associated with a name. You may include text indirectly in at least three situations:

- If the text includes characters not found on your keyboard. For example, the symbols alpha (α), pi (π), or delta (δ) are not on most keyboards. But you can include such characters in your overlay by specifying **HEX** in the **SETTEXT** or **WITHTEXT** commands.
- If the text is used repeatedly on the form or is commonly used by your organization, for example, the confidentiality statement on the sample overlay “PREREG”.
- If the text is long and you do not want it to appear in the overlay definition, for example, a statement explaining copyright restrictions.

Your system programmer can tell you if a symbolic data set exists or how to create one. See Appendix E, “The Symbolic Data Set and Symbolic File” on page 319 for more information.

A symbolic data set, which consists of fixed-length 80-byte records and can include one or more text strings, contains the following for each string:

CONF1	C 'Information on this form is strictly confidential'	
CONF2	C ' and will not be released without your consent.'	
name	text	text
	type	

Figure 109. Symbolic Data Set or File Records

- name* This name is the name of the text string in the symbolic data set or file record. This name is used in the **SETTEXT** command or the **WITHTEXT** subcommand to identify the text. The only limit to the length of the name is that the record, the text type, and the text cannot exceed 80 bytes. The name must begin in column 1 of the record.
- text type* The text type is separated from the name by one or more blanks. Text type is one of the following:
- C** For character text.
 - X** For hexadecimal text.
- For more information on the text types in the symbolic data set or file, see “Symbolic Data Sets and Symbolic Files Containing Double-Byte Characters” on page 320.
- text* The text, enclosed in apostrophes, immediately follows the text type (no spaces).

Figure 109 illustrates records in the symbolic data set or file, that define the confidentiality statement in the sample overlay.

To use the text contained in the record of a symbolic data set, observe the following rules (for double-byte fonts, such as Kanji fonts, see “Symbolic Data Sets and Symbolic Files Containing Double-Byte Characters” on page 320):

- The name, preceded by an ampersand and followed by a period, is included in the text entry of **SETTEXT** or **WITHTEXT**. The text itself, whether entered directly or as a symbolic name, must be enclosed in apostrophes.
- A single text entry can include one or more symbolic names and can also include text entered directly.

Text can be specified in a **SETTEXT** command or in a **WITHTEXT** subcommand in two ways:

Directly: Enter the actual characters or hexadecimal codes for the characters.

Indirectly: Enter the name that identifies the string of text stored in the symbolic data set or file. The name becomes the “symbolic” for the text.

The command to include the confidentiality statement on the sample overlay is:

```
SETTEXT 0 MODERN LEFT SPACED 1
      LINE FONT5 NOUNDERLINE CHAR '&CONF1'
      LINE FONT5 NOUNDERLINE CHAR '&CONF2';
```

When using symbolic data sets:

- A space between two symbolic names results in a space being printed between two text segments. For example, if you want the entire confidentiality statement to print on one line, you enter:

```
LINE FONT5 NOUNDERLINE CHAR '&CONF1 &CONF2';
```

- In the **SETTEXT** command or the **WITHTEXT** subcommand, the keyword **CHAR** or **HEX** refers to the contents of the text in that entry, not the contents of the symbolic data set or file. In the example above, **CHAR** is the correct word whether the contents of the text in the symbolic data set or file is in character form or hexadecimal form.

Symbolic Data Sets or Files

Chapter 5. Adding Graphics

In OGL/370, there are two kinds of graphics:

Segments: Graphics that have already been created and stored in a library (just as the fonts have been). To use these graphics, you must know the member name of the segment. Ask your system programmer for this information.

Image patterns: Graphics that you design as part of the overlay definition.

The college seal (the palm tree) and the president's signature on the sample overlay are segments. The initials of the college (TCC) are patterns.

Note: An important difference between segments and patterns is that OGL/370 is used to define patterns, but not to modify segments. These modifications are explained in the description of the **PLACE** command. See "Placing Graphics (**PLACE**)" on page 125 for more information.

Choosing Graphics (SEGMENT)

You tell the printer to access a segment the same way you tell it what fonts to use. As with fonts, you must identify the segment in your definition before you do anything with it.

See your system programmer for the following information:

- What segments are available?
- What do the segments look like (including size and orientation)?
- What are their member names (MVS and VSE) or filenames (VM)?
- What is the name of the DD statement (MVS) that identifies the segment data set?
- What is the filetype (VM) of the segment?

Write a separate **SEGMENT** command for each segment chosen.

Note: Each system uses different **SEGMENT** command entries. See "**SEGMENT** Command (MVS)" on page 283., "**SEGMENT** Command (VSE)" on page 285, or "**SEGMENT** Command (VM)" on page 284 for more information. The parts of the **SEGMENT** command are as follows:

SEGMENT	palm	palm2	<u>DDNAME</u>	<u>SEGDD</u>	;
command	segment	member	dd stmt	dd stmt	end
word	name	ID	name	word	marker

Figure 110. **SEGMENT** Command for MVS. For the **VSE SEGMENT** or **VM SEGMENT** command, you only need to specify the command word, segment name, and member ID.

command word

SEGMENT

segment name

This entry, which is optional, is the name you use to place the segment. If you do not specify a name in the **SEGMENT** command for this segment, OGL/370 uses the *member ID* (the next entry in this command) as the name, and you must use this name in the **PLACE** command to place the segment. The segment name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Choosing Graphics

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

member ID This ID, which cannot be more than six characters long, is taken from the member name under which the segment is stored. The only difference is that the member name has a two-character prefix, "S1". For example, the palm-tree segment is stored under the member name "S1PALM2". In the **SEGMENT** command, you enter "PALM2" for the *member ID*.

DD statement name (MVS only)

Enter the name of the DD statement on the JCL that names the segment library used.

Ask your system programmer for this information. The default is **DDNAME**. If you are using the standard name, enter either **DDNAME** or nothing.

DD statement word (MVS only)

SEGDD (Default)

end marker Always end a command with an end marker (;).

Along with OGL/370, your company received two segments, the college seal (S1PALM2) and the signature (S1SIGNAT). Figure 110 on page 117 contains the **SEGMENT** command for the college seal, which we have named "PALM".

Designing Graphics (DEFINE)

To draw a graphic, the printer must know which pels to tone and which to leave untone. Because there are 240 pels per inch, an image 1-inch square consists of a pattern of 57,600 pels. Using OGL/370, you can code each pel individually or code each line of pels. For a 1-inch square image, you need 240 lines of pel coding.

Look at overlay "RCPT" (Figure 14 on page 23). To the right of the seal is an image of the college initials. This image is relatively small and simple. But if you treat the initials as two images, one large "T" and one small "C" (used twice), you simplify the coding and save printer storage space.

In Figure 111, Part A shows the initials as one image and requires 10,260 pels. Part B involves two images and requires only 8,000 pels.

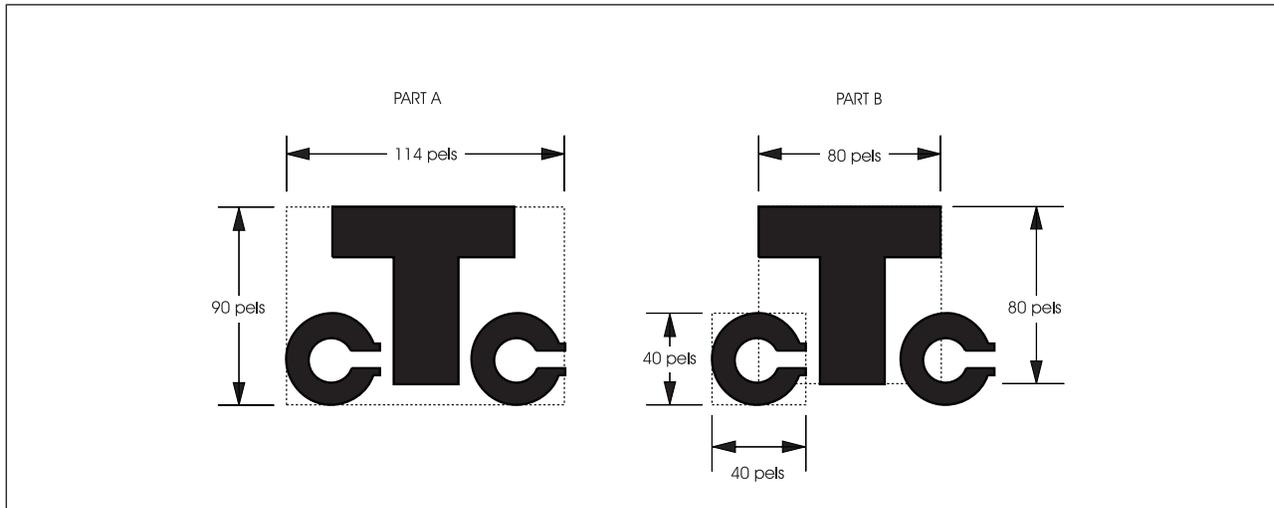


Figure 111. Two Ways of Looking at the Same Image. (Not actual size.)

Before you write the **DEFINE** command, draw an enlarged version of the image on a mask in which each square represents a pel.

Note: If you are using OGL/370 under MVS, IBM supplies a job, DZISAMP3, which is loaded as a member in SYS1.SAMPLIB. To produce the sample overlay shown in Figure 112 on page 120, you must run the DZISAMP3 job. You can print copies of the mask as you would any overlay. Because of the size of the pel form, it should be printed on paper at least 14.875 inches wide by 11 inches high.

Designing Graphics

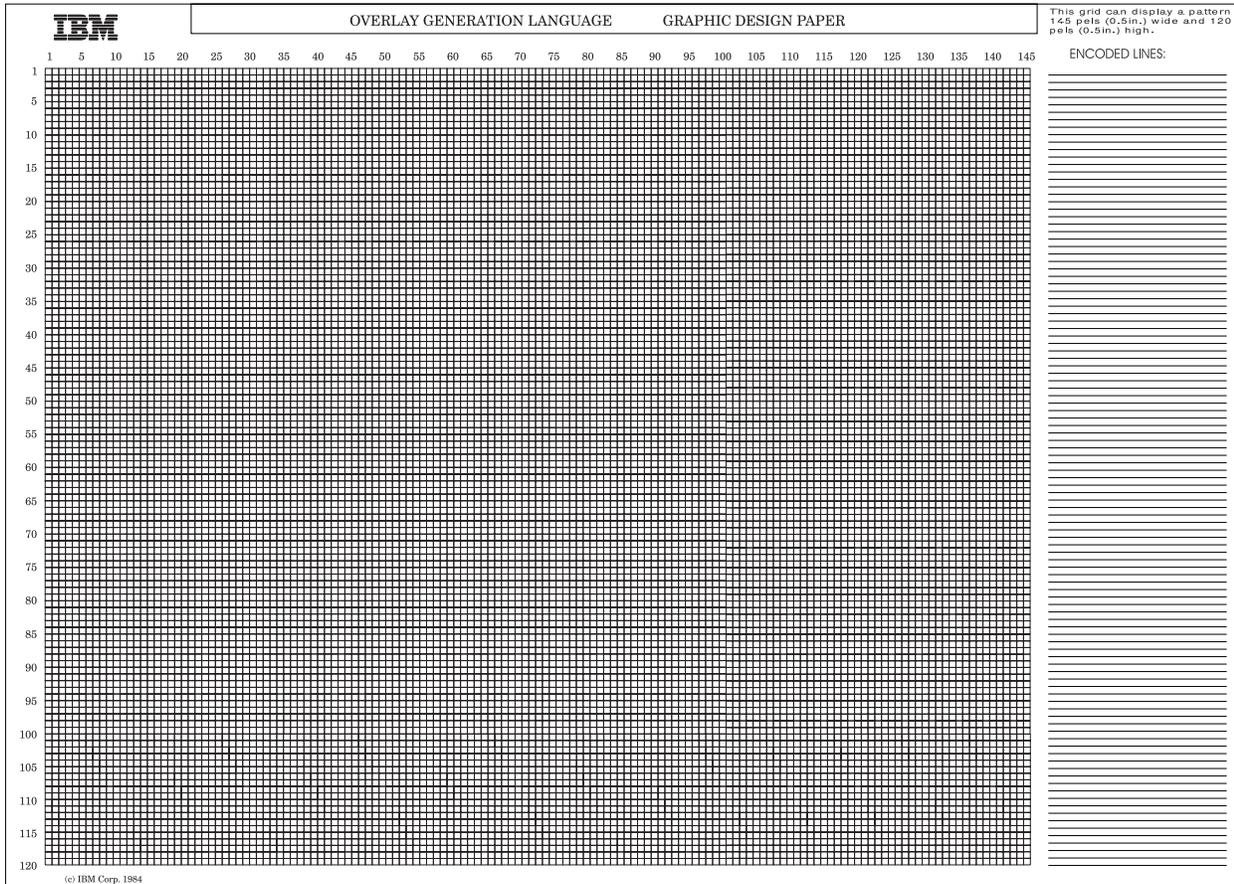


Figure 112. Overlay Mask for Designing Graphics. The actual size of the overlay is 13.8 inches by 10.0 inches.

After you draw the image, you are ready to write the **DEFINE** command.

The parts of the **DEFINE** command are as follows:

DEFINE	bigt	PATTERN	ENCODED	(0 80)	
				.	
				.	
				.	
				(27 26)	;
command	section	section	pattern	line	end
word	name	type	type	coding	marker

Figure 113. **DEFINE** Command

command word

DEFINE

section name This is the name you use to place the image, which is a type of overlay section. The section name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

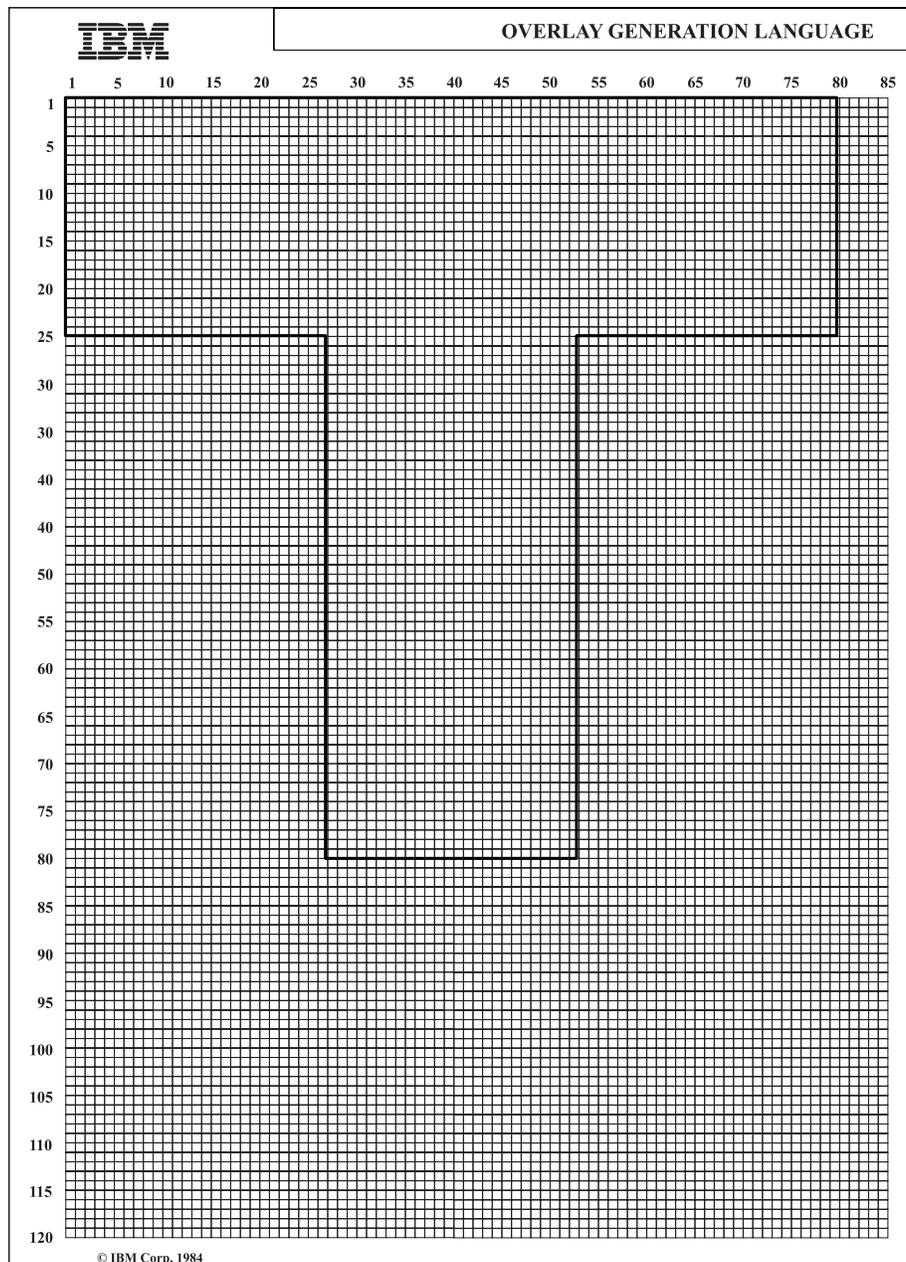


Figure 114. Pel Pattern for the "T"

Lines 1 through 25 look like this if **ENCODED**:

(0 80)

Note: Although the first pel is to be toned, you must specify the number of initial untoned pels (in this case, 0).

Lines 26 through 80 look like this:

(27 26 27)

Note: Untoned pels that are not followed by toned pels do not need to be coded. The above line is coded simply as:

(27 26)

end marker Always end a command with an end marker (;).

Note: The end marker is written only at the end of the entire command.

Figure 113 on page 120 shows the **DEFINE** command for the “T” image. Only the first and the last lines of pattern coding are shown; the ellipses represent the lines not displayed.

The command used to define the “C” is similar to the one used to define the “T”. However, the pel pattern shown in Figure 115 is more complicated.

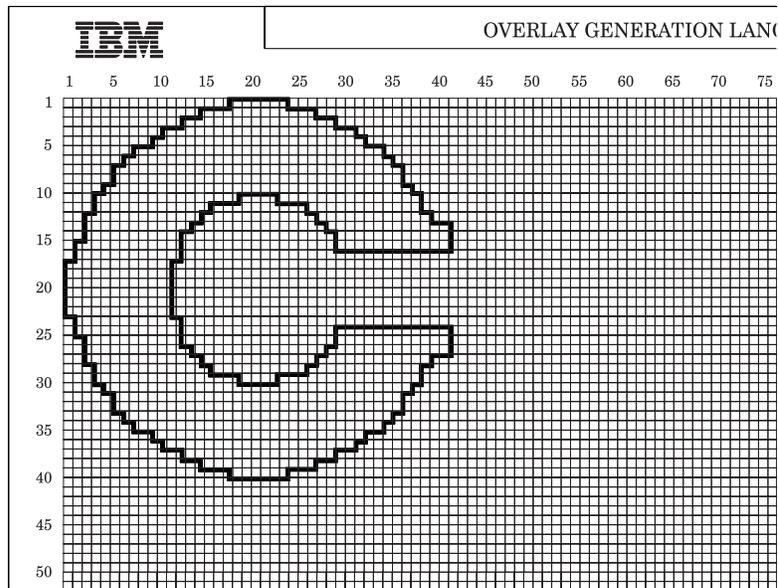


Figure 115. Pel Pattern for the “C”

The edges look rough, but the printed image is much smaller than this illustration. The complete **DEFINE** command for the “C” pattern is shown in Figure 116 on page 124. The numbers in the right column identify the line of pels.

Designing Graphics

```
-'C PATTERN'  
  DEFINE SMALLC PATTERN ENCODED (17 6)      -1  
                                  (14 12)    -2  
                                  (12 16)    -3  
                                  (10 20)    -4  
                                  (9 22)     -5  
                                  (7 26)     -6  
                                  (6 28)     -7  
                                  (5 30)     -8  
                                  (5 30)     -9  
                                  (4 32)    -10  
                                  (3 15 4 15) -11  
                                  (3 12 10 12) -12  
                                  (2 12 12 12) -13  
                                  (2 11 14 13) -14  
                                  (2 10 16 12) -15  
                                  (1 11 16 12) -16  
                                  (1 11)     -17  
                                  (0 11)     -18  
                                  (0 11)     -19  
                                  (0 11)     -20  
                                  (0 11)     -21  
                                  (0 11)     -22  
                                  (0 11)     -23  
                                  (1 11)     -24  
                                  (1 11 16 12) -25  
                                  (2 10 16 12) -26  
                                  (2 11 14 13) -27  
                                  (2 12 12 12) -28  
                                  (3 12 10 12) -29  
                                  (3 15 4 15) -30  
                                  (4 32)     -31  
                                  (5 30)     -32  
                                  (5 30)     -33  
                                  (6 28)     -34  
                                  (7 26)     -35  
                                  (9 22)     -36  
                                  (10 20)    -37  
                                  (12 16)    -38  
                                  (14 12)    -39  
                                  (17 6)     -40;
```

Figure 116. The **DEFINE** Command for the “C” Pattern

When OGL/370 processes an image, it uses untuned pels to round out the image-pattern height and width to multiples of 8, unless the images are already defined as multiples of 8, as are the “T” and the “C”.

Thus, if you define SMALLC as 33 pels wide and 33 pels high, OGL/370 creates an invisible box 40 pels by 40 pels. What is printed, however, is what you asked for.

You must take this invisible box into consideration if you position an image close to the edge of the overlay. For example, if you position the “C” 33 pels from the overlay side or bottom, it is not printed. OGL/370 sees an image box that extends 7 pels over the side and 7 pels below the bottom of the overlay.

Positioning Graphics (POSITION)

Overlays and boxes are positioned by their top-left corners (origins). The same is true of graphics. The printer sees all graphics as rectangles, even if they are invisible.

Figure 117 illustrates how the printer sees the college seal, the signature, and the letter “T”. The dotted rules represent invisible boxes.



Figure 117. Invisible Boxes around Graphics

Placing Graphics (PLACE)

To print a box, write a **POSITION** command and define the box with the **DRAWBOX** command. Graphics are placed on a page differently, you must:

1. Define the image pattern (**DEFINE**) or identify the segment (**SEGMENT**).
2. Write the **POSITION** command (it need not immediately follow the **DEFINE** or **SEGMENT** commands).
3. Write the **PLACE** command after the **POSITION** command to name the graphic you want placed.

The parts of the **PLACE** command used to place a segment are as follows:

PLACE	SEGID	palm	;
command word	section type	section name	end marker

Figure 118. **PLACE** Segment

If you want to place the same graphic more than once on an overlay (as we do with the “C”), you do not have to repeat the **SEGMENT** or the **DEFINE** commands. Each time you use the **POSITION** and the **PLACE** commands, the printer prints the graphic.

The **PLACE** command with pattern modifications:

PLACE	PATTERN	bigt	<u>0</u>	SHADE	STANDARD	LIGHT	NOMIRROR	NONEGATIVE	;
command word	section type	section name	orien- tation	shading option	shade pattern	shade type	mirror option	negative option	end marker

Figure 119. **PLACE** Pattern

command word

PLACE

section type

Choose from:

SEGID

PATTERN

Placing Graphics

section name The name given to the segment or pattern in the **SEGMENT** or **DEFINE** command. If you did not assign a name to the segment, use the *member ID*. See page 201 for naming requirements.

Pattern Modifications: The following options apply only to patterns.

orientation The number you choose defines the orientation of the pattern relative to the overlay. Choose from:
0 (Default)
90
180
270

Figure 120. illustrates four orientations for a pattern of the Greek letter “sigma” (σ).

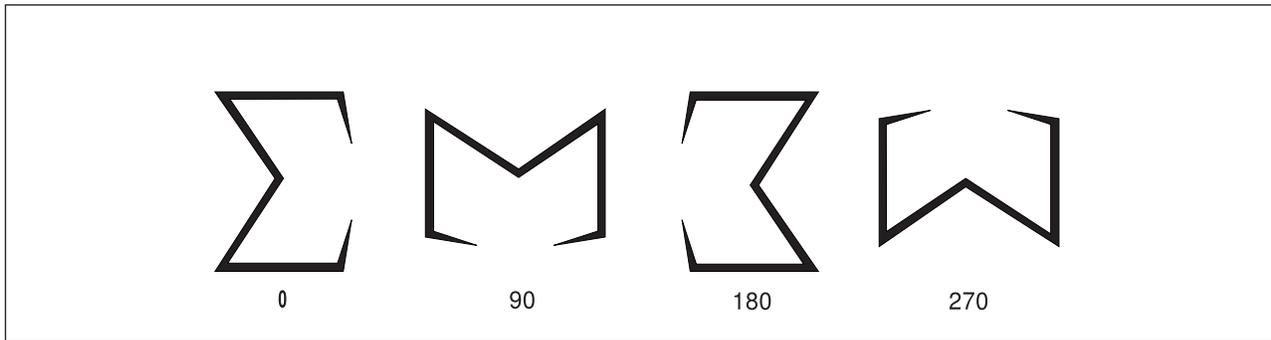


Figure 120. Pattern Orientation

shading option **SHADE**

Choosing this option indicates that you want the pattern shaded according to the following patterns and types.

Note: If the lines of the original image are very fine (1 or 2 pels), using the shading option may obliterate some lines or may render some or all of the image unclear.

shade pattern Choose the shade pattern you want from the examples in Figure 212 on page 330.

STANDARD (Default)
SCREEN

shade type This name or number specifies the shade type that you want applied to the pattern.

XLIGHT
LIGHT
MEDIUM (Default)
DARK
XDARK

n A percentage of shading, do not add the percent symbol (%).

Figure 121 on page 127 shows the “sigma” (σ) pattern printed in the five named shades. As with boxes, you can specify the shading percentage. Figure 121 on page 127 shows the named shades. Figure 211 on page 326 shows the shading percentages.



Figure 121. Pattern Shading

mirror option A pattern can be printed as you defined it, or its mirror image can be printed, depending on which of the following you specify:

NOMIRROR (Default)

MIRROR

Figure 122 shows the “sigma” (Σ) pattern specified with **MIRROR**.

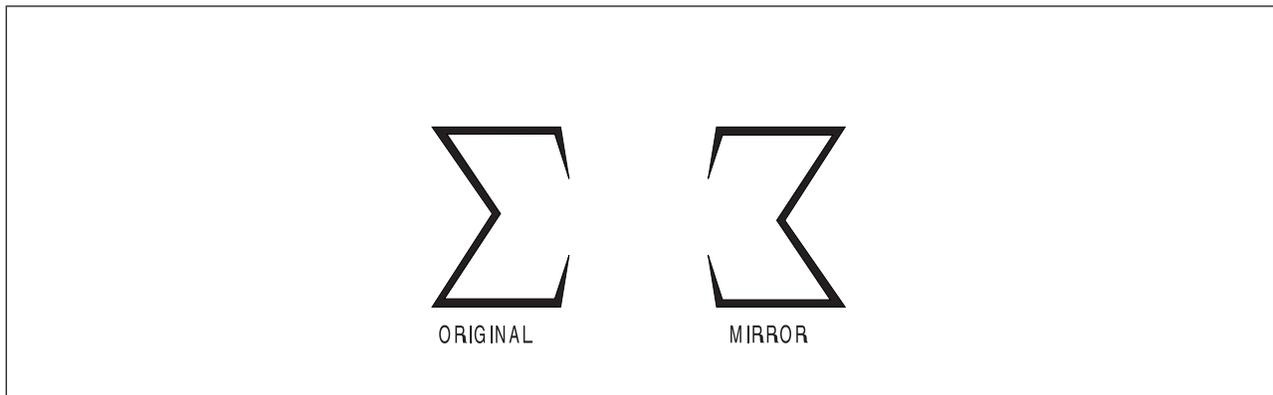


Figure 122. **MIRROR** Option

negative option

A pattern can be printed as you defined it or its negative image patterns can be printed, depending on which of the following you specify:

NONEGATIVE (Default)

NEGATIVE

Figure 123 on page 128 shows the “sigma” (Σ) pattern specified with **NEGATIVE**.

Placing Graphics

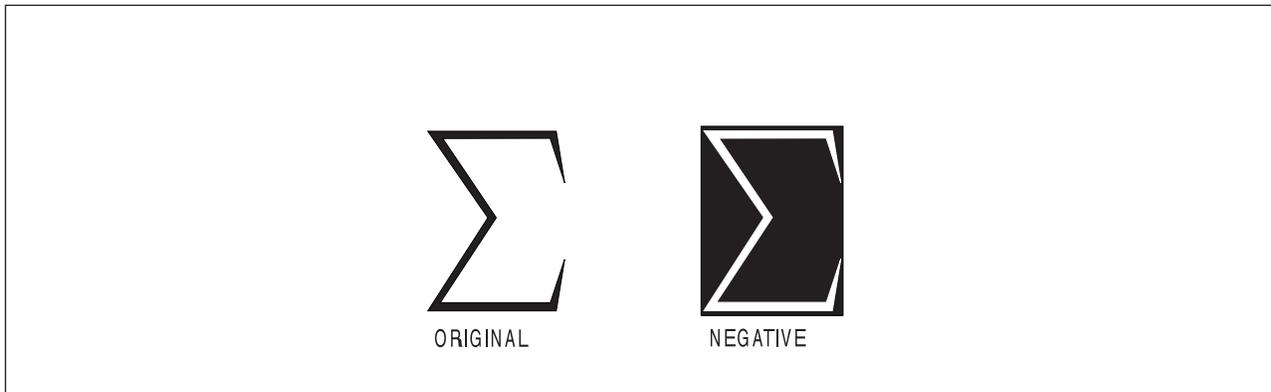


Figure 123. **NEGATIVE** Option

Now you can write the commands needed to draw the college seal, the president's signature, and the "T" at the correct places on the overlay.

```
- 'ADDING GRAPHICS'  
  SEGMENT PALM PALM2 DDNAME SEGDD;  
  SEGMENT PRES SIGNAT DDNAME SEGDD;  
  
  POSITION ABSOLUTE .25 IN ABSOLUTE 1.75 IN;  
  PLACE SEGID PALM;  
  
  POSITION ABSOLUTE 3.4 IN ABSOLUTE 2.32 IN;  
  PLACE SEGID PRES;  
  
  DEFINE BIGT PATTERN ENCODED (0 80)          -1  
                                .  
                                .  
                                (27 26)      -80  
  POSITION ABSOLUTE 1.95 IN ABSOLUTE 2.4 IN;  
  PLACE PATTERN BIGT 0 NOMIRROR NONEGATIVE;
```

Figure 124. *Commands for Drawing Graphics (MVS).* For VSE or VM, your **SEGMENT** command requires only the command word, segment name, and member ID entries.

The order of your commands may be slightly different. It is correct to write all the **SEGMENT** and **DEFINE** commands together and later write the **POSITION** and **PLACE** commands.

Remember that patterns can be rotated and segments cannot. Figure 125 on page 129 shows four different segments, each 136 pels wide by 232 pels high. Each segment is a different orientation of the "sigma" (Σ) pattern. We also have one pattern definition of the same graphic in the 0° orientation.

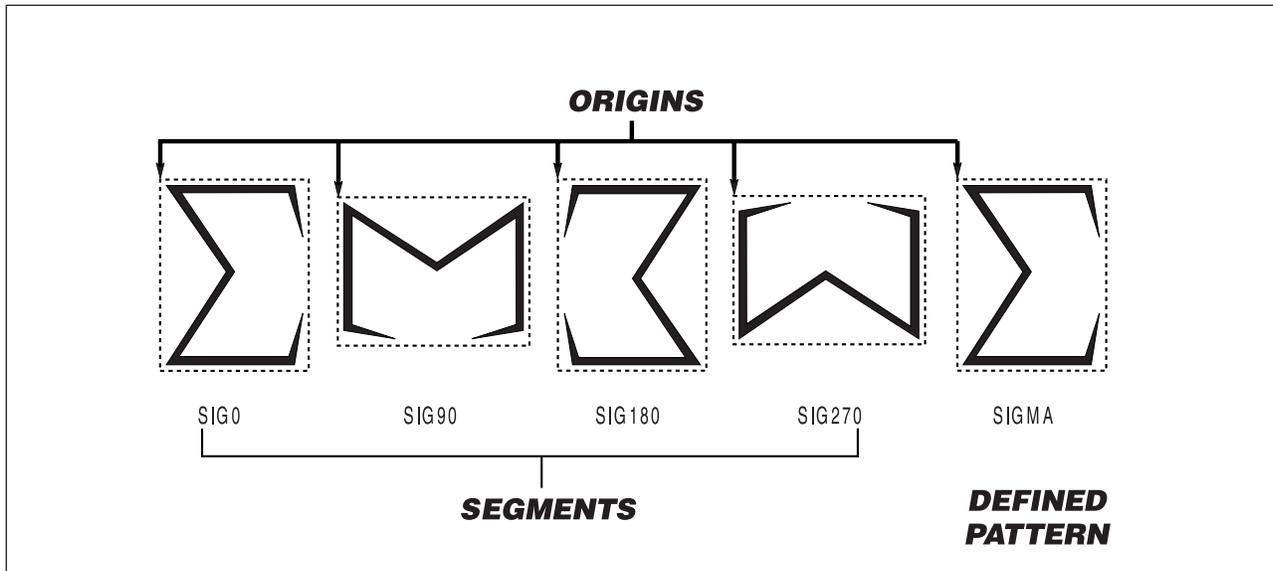


Figure 125. Graphic Origins

The “**PATTERN**” illustration in Figure 126 shows what happens when a pattern is rotated around a single point. The commands that produce this result are also shown. The “**SEGMENTS**” illustration shows the same result using segments and the required commands. If you use the segments, you must use a **POSITION** and **PLACE** pair for each segment as shown. However, if you use the one pattern, you write one **POSITION** command and four **PLACE** commands, each specifying a different orientation.

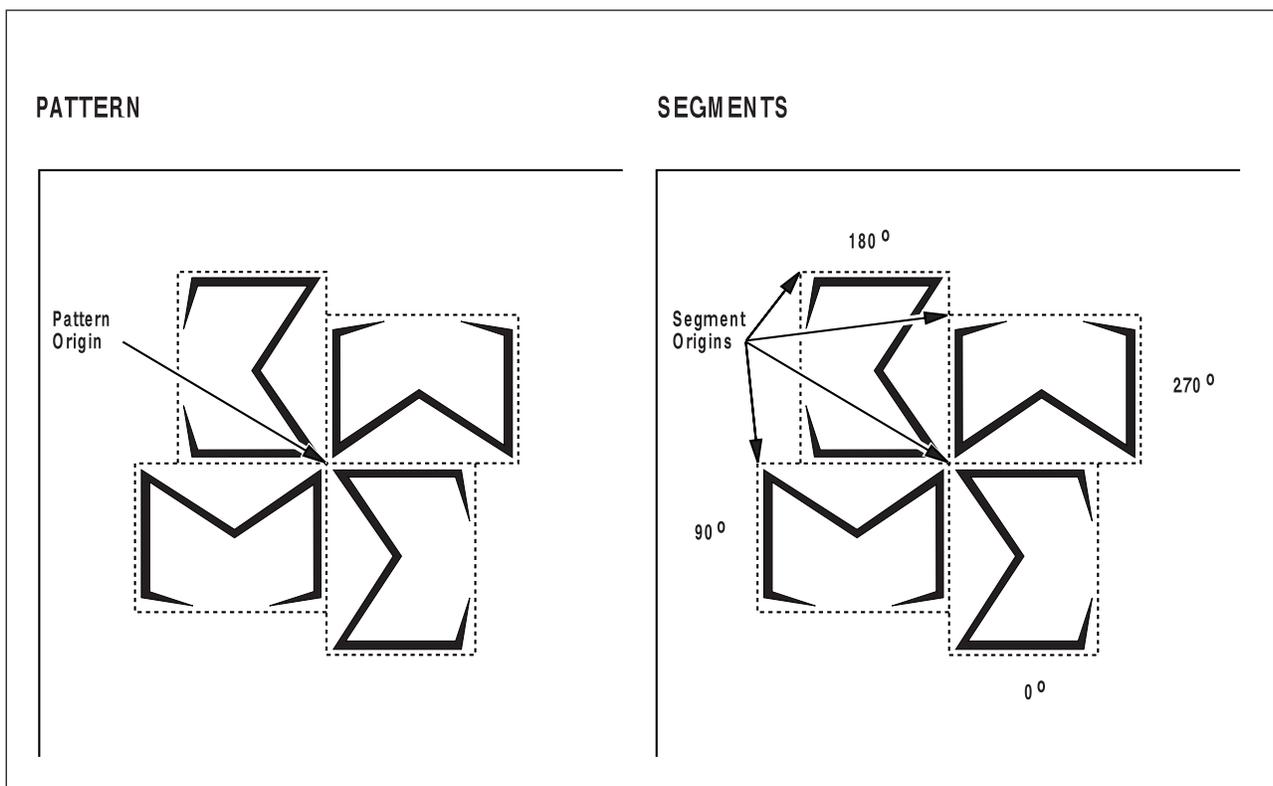


Figure 126. Graphics in Different Orientations and the Same Positions

Placing Graphics

Table 1. OGL/370 commands for Graphics in Different Orientations

Pattern Origin:	Segment Origin:
POSITION ABSOLUTE 1.5 IN ABSOLUTE 1.5 IN; PLACE PATTERN SIGMA 0; PLACE PATTERN SIGMA 90; PLACE PATTERN SIGMA 180; PLACE PATTERN SIGMA 270;	POSITION ABSOLUTE 1.5 IN ABSOLUTE 1.5 IN; PLACE SEGID SIGM0; POSITION ABSOLUTE 129 PELS ABSOLUTE 1.5 IN; PLACE SEGID SIGM90; POSITION ABSOLUTE 225 PELS ABSOLUTE 129 PELS; PLACE SEGID SIGM180; POSITION ABSOLUTE 1.5 IN ABSOLUTE 225 PELS; PLACE SEGID SIGM270;

Chapter 6. Adding Color

To color text, box rules, **DRAWRULEs**, box background space, and pattern image, you must first create a color definition, then call out that color definition on a subset of OGL commands.

Use **DEFINE COLOR** to create a color definition using the RGB, CMYK, IBM OCA, or IBM Highlight color models.

IBM OCA

This color model allows you to use the following predefined printer-specific colors: blue, red, magenta, green, cyan, yellow, black, brown, darkblue, orange, purple, darkgreen, darkcyan, mustard, gray, none, and default. The color *none* means no other color is applied.

RGB

This color model allows you to specify the percent of red, green, and blue to create your color. This is called process color. You describe the color rather than picking a predefined named color. The percentages given specify how much of each color to use. The percentages do not have to add up to 100. For example, the following defines a color that contains 100% blue, no red and no green:

```
define blue1 color rgb rval 0 gval 0 bval 100 ;
```

CMYK

This color model allows you to specify the percentage of cyan, magenta, yellow, and black that compose your color. This is also a process color model. The percentages do not have to add up to 100. The following defines a color that contains 100% cyan, no magenta, no yellow, and no black:

```
define gree1 color cmyk  
  cval 100 mval 0 yval 100 kval 0;
```

Highlight

This color model allows you to specify a particular device-dependent color such as that used on the IBM InfoPrint Hi-Lite Color Post Processor, the percentage of that color to use and the percentage of black to add to the color. The color assigned to each highlight color is determined by your output device. The percentages given cannot add up to more than 100. If they add up to less than 100, the rest of the color will be made up from the color of the medium you are printing on. The following defines a color as color 2, using 60% of the color, 20% black, and 20% color of medium.

```
define high2 color highlight 2 coverage 60 black 20;
```

Cielab

This color model allows you to define a color with three-dimensional geometry using 1 luminance and 2 chrominance values. The luminance value can be specified as a percentage up to 2 decimal places. The chrominance values are specified as integers in the range -127 to 127:

```
define cie2 color cielab lval 10 c1val(-50) c2val(5);
```

Box Background Color

On the **DRAWBOX** command, you can specify a color definition to be used for the box background space.

Foreground Color

You can associate a color with the following OGL objects:

- The font used for **SETTEXT** or **WITHTEXT** text
- The box border rules on **DRAWBOX**
- The **DRAWRULE** and **DRAWMASK** rules
- The on pels when you place a pattern on the page.

Adding Color

Putting it All Together

The following VM example shows definitions for all five color models and how to call out those definitions on the various OGL commands:

```
CONTROL NOSTORE nosummary;
OVERLAY xxx SIZE 8.5 IN 11 IN OFFSET 0 0 ;
ORIENT 0;
define red1 color oca red ; /* predefined IBM color */
define blue1 color rgb rval 0 gval 0 bval 100 ; /* % of red green blue */
define gree1 color cmyk cval 100 mval 0 yval 100 kval 0; /* % of cyan magenta yellow black */
define cie1 color cielab lval 10 clval(50) c2val(10); /* luminance and chrominance values */
define cie2 color cielab lval 10 clval(120) c2val(-5); /* luminance and chrominance values */
define nocol color oca none ; /* predefined IBM color */
define high1 color highlight 1 coverage 50 ; /* highlight color */

font font1 n2200e filetype fontoln -'outline fonts'
height 14 scale 80 color blue1 ucolor cie1;
font font2 n2200e filetype fontoln
height 14 color nocol;

drawmask 1 in 1 in hcolor gree1 vcolor blue1 ;

position .5 in .5 in; -'text with color'
settext modern left line font1 underline
'This is RGB blue text with cielab underscore' ;

position 1 in .75 in ; -'repeat box'
drawbox 3 in 1 in bold dashed
bcolor red1 diagonal left
color box 1 blue1
box 2 cie1
box 3 cie2
box 4 gree1
box 5 high1
repeat down 4 spaced 2 MM
withtext box 1 line font2 'OCA red dashed box rules'
line font2 'RGB blue box background'
withtext box 2 line font2 'OCA red dashed box rules'
line font2 'CIE1 box background'
withtext box 3 line font2 'OCA red dashed box rules'
line font2 'CIE2 box background'
withtext box 4 line font2 'OCA red dashed box rules'
line font2 'CMYK green box background'
withtext box 5 line font2 'OCA red dashed box rules'
line font2 'Highlight 50 box background' ;
position 6 in .75 in ; -'OCA red vertical rules'
drawrule down 3 in bold
repeat across 3 spaced 4 MM
color red1;
```

The following is an example of highlight color for the IBM InfoPrint Hi-Lite Color Post Processor:

```
-' Example for InfoPrint Hi-Lite Color post processor
CONTROL NOSTORE ALL noSUMMARY
OVERLAY xmp2 SIZE 8.5 IN 11 IN OFFSET 0 IN 0 IN;
ORIENT 0;

DEFINE h00      COLOR HIGHLIGHT 0 COVERAGE 0 ;
DEFINE high020  COLOR HIGHLIGHT 0 COVERAGE 20;
DEFINE high130  COLOR HIGHLIGHT 1 COVERAGE 30;
DEFINE high240  COLOR HIGHLIGHT 2 COVERAGE 40;
DEFINE high3100 COLOR HIGHLIGHT 3 COVERAGE 100;
DEFINE high0    COLOR HIGHLIGHT 0 ;
DEFINE high1    COLOR HIGHLIGHT 1 ;
DEFINE high2    COLOR HIGHLIGHT 2 ;
DEFINE high3    COLOR HIGHLIGHT 3 ;

FONT FONT1     GT15 FILETYPE FONT3820 ;
FONT fonth0    GT15 FILETYPE FONT3820 color high0;
FONT fonth1    GT15 FILETYPE FONT3820 color high1;
FONT fonth2    GT15 FILETYPE FONT3820
               color high2 ucolor high3;
FONT fonth3    GT15 FILETYPE FONT3820 color high3;
FONT f00       GT15 FILETYPE FONT3820 color h00 ;

POSITION .25 IN .25 IN;
settext modern left line font1 'Testcase HILITXMP';
POSITION .25 IN .5 IN;
settext modern left
  line font1 'Dev Def Color'
  line fonth2 underline 'color 2 with col 3 underscore'
  line fonth0 'device default color'
  line fonth2 'color 1'
  line fonth2 'color 2'
  line fonth3 'color 3' ;
POSITION .25 IN 1.5 IN;
settext modern left
  line font1 'box rules=color 2' ;
POSITION .25 IN 1.75 IN;
DRAWBOX 2.5 IN 1 IN medium bcolor high2
repeat down 3 spaced 2 mm
  color box 1 high020
  box 2 high130
  box 3 high240
  box 4 high3100
withtext box 1
  line fonth3 'background=color 0, 20%'
  line fonth3 'text=color 3 at 100%'
withtext box 2
  line fonth2 'background=color 1, 30%'
  line fonth2 'text=color 2 at 100%'
withtext box 3
  line fonth1 'background=color 2, 40%'
  line fonth1 'text=color 1 at 100%'
withtext box 4
  line fonth1 'background=color 3, 100%'
  line fonth1 'text=color 0 at 0%' ;
```

Adding Color

Part 3. Additional Features

Chapter 7. Additional Features and Commands

Now that you have finished Part 2, “Designing Overlays” on page 21, you can write definitions for overlays with boxes, rules, circles, paths, pictures, and text.

This chapter describes features of the **DRAWRULE** and **DRAWBOX** commands. It also introduces the **SETUNITS** command, which allows you to specify default values for units of measurement and other aspects of overlay definition.

These features are presented in five sections:

- “Using Default Options (**SETUNITS**)” explains how to specify a unit of measurement as a default. It also describes how to specify other default values that can be used when defining your overlay.
- “Repeating Rules (**DRAWRULE REPEAT**)” on page 149 explains how to repeat a rule by using the **REPEAT** subcommand in the **DRAWRULE** command.
- “Repeating Boxes (**DRAWBOX REPEAT**)” on page 154 explains how to repeat a box by using the **REPEAT** subcommand in the **DRAWBOX** command.
- “Defining and Placing Groups” on page 162 explains how to define an entire section of an overlay as a named group and place it on the overlay wherever and as often as you choose.
- “More Features of **DRAWBOX**” on page 167 explains how to specify boxes to have features such as rounded corners, diagonals and dotted or dashed borders. It also describes how to specify balancing and justification of text.

The first section in this chapter presents the **SETUNITS** command, which sets up defaults for horizontal and vertical measurements, as well as other default values. Because all the other sections assume you are familiar with this command, you should read “Using Default Options (**SETUNITS**)” first. Once you are familiar with **SETUNITS**, you can read the other sections in any order you want, although you might find it easier to follow the order presented in this chapter.

Using Default Options (**SETUNITS**)

This section tells you about the functions of the **SETUNITS** command.

- “General Features of **SETUNITS**” on page 138 contains general facts about the way **SETUNITS** works, such as how several **SETUNITS** commands interact.
- “Default Units of Measurement” on page 138 explains how to use **SETUNITS** to set a default unit of measurement for horizontal and vertical distances.
- “**SETUNITS** Example” on page 139 gives an example of using the various features of **SETUNITS**.
- “Line Spacing with **SETUNITS**” on page 141 explains how to use **SETUNITS** to set default line spacing for text strings.
- “Corner Length Values with **SETUNITS**” on page 141 explains how to use the **CORNERLENGTH** subcommand to set the degree of rounding for boxes with rounded corners and the default rounding for paths with rounded connections.
- “Text Margins with **SETUNITS**” on page 144 explains how you use the **TEXTMARGIN** subcommand to specify the way text strings should be placed within a box or circle.
- “Top-Left and Center Positioning with **SETUNITS**” on page 144 explains how to use the **POSITIONING** subcommand to determine whether rules and boxes should be positioned using top-left or center positioning.
- “**SETUNITS** and **DRAWMASK**” on page 147 explains how to use **SETUNITS** to set the line spacing values of a mask created by the **DRAWMASK** command.

Using Default Options

General Features of SETUNITS

Use the **SETUNITS** command as often as you wish. In general, any option you set with **SETUNITS** (such as a default unit of measurement or box and rule positioning) stays in effect until you change it by specifying the option again in another **SETUNITS** command. See “Using **SETUNITS** inside a Group (**SETUNITS**)” on page 164 for special considerations when using **SETUNITS** within a group.

Default Units of Measurement

In many commands, you specify one or more dimensions by entering a number and a unit of measurement, for example:

```
DRAWBOX 2 IN 1 IN;
```

Using the **SETUNITS** command, you can define horizontal and vertical units of measurement that can be used as the default units in other commands. For example, if the **SETUNITS** command establishes 1 inch as the horizontal and vertical default unit, the preceding **DRAWBOX** command might look like this:

```
DRAWBOX 2 1;
```

OGL/370 uses 1 inch for the unit of measurement and multiplies it by 2 for the width of the box and by 1 for the height of the box. The result is a box 2 inches wide and 1 inch high. On the other hand, if the **SETUNITS** command establishes 2 inches as the horizontal and the vertical default unit, what would the previous **DRAWBOX** command produce? The unit of measurement, 2 inches, is multiplied by 2 for the box width and 1 for the box height, resulting in a box 4 inches wide and 2 inches high.

Using default measurement units can save you time while you write the overlay definition. But the command is more useful than that. Chapter 2, “Designing a Simple Overlay” on page 23 describes the design of a sample overlay on a mask of grid rules at $\frac{1}{4}$ -inch intervals, with most of the measurements (that is, sizes and positions) in multiples of $\frac{1}{4}$ -inch.

If we defined the default unit of measurement as 0.25 inches, how can you write the following pair of commands?

```
POSITION ABSOLUTE 1 IN ABSOLUTE .5 IN;  
DRAWBOX .75 IN 1 IN;
```

The **POSITION** command specifies a move 1 inch across (four intervals of $\frac{1}{4}$ -inch each) and $\frac{1}{2}$ -inch down (two intervals). Then the **DRAWBOX** command describes a box $\frac{3}{4}$ -inch wide (three intervals) and 1 inch high (four intervals). These are the rewritten commands:

```
POSITION ABSOLUTE 4 ABSOLUTE 2;  
DRAWBOX 3 4;
```

Consider the case of an overlay designed on a $\frac{1}{4}$ -inch grid. For example, if you think a box would look better if it were half an interval wider ($\frac{3}{8}$ inch total), you would not have to figure out its decimal equivalent (0.375 inches). Using the **SETUNITS** default, specify the width as 1.5.

The **SETUNITS** command can establish defaults in the following units:

- Fractions ($\frac{1}{4}$), multiples of an inch (.30 **IN**), or millimeters (2 **MM**)
- One or more pels (24 **PELS**)
- Characters per inch (6 **CPI**)
- Lines per inch (5 **LPI**).

You can mix default values. For example, you can set the horizontal default to 0.10 inches and the vertical default to 5 millimeters.

With 8 **CPI** and 12 **LPI** set as horizontal and vertical defaults, respectively, what are the width and height of the following box?

```
DRAWBOX 4 6;
```

The box is $\frac{1}{2}$ inch wide and $\frac{1}{2}$ inch high.

As you can see, **LPI** and **CPI** work in exactly the same way. The only difference is that you may want to use **LPI** for text printed in the **0°** and **180°** orientations and **CPI** for text printed in the **90°** and **270°** orientations.

In any command that calls for a unit of measurement, you can override the **SETUNITS** default unit by specifying the unit (for example, **IN** or **MM**) explicitly.

SETUNITS Example

Figure 127 shows an example of using the **SETUNITS** command. The primary and secondary defaults have been described already. The other options are introduced briefly in this section and explained in more depth in separate sections.

The **SETUNITS** command contains the following elements:

SETUNITS	.25 in	.25 in	LINESP	.15 in
command word	primary default	secondary default	spacing option	spacing value
CORNERLENGTH	MAX	TEXTMARGIN	SQUARE	
subcommand word	length value	subcommand word	text margin	
POSITIONING	CENTER	;		
subcommand word	positioning option	end marker		

Figure 127. **SETUNITS** Command

command word

SETUNITS

primary default

If both the primary and secondary defaults are specified, the primary default applies only to horizontal measurements. If the secondary default is not specified, the primary default applies to both horizontal and vertical measurements. Specify a number (*n*) and one of the following units of measurement:

Using Default Options

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch
CPI	Characters per inch

secondary default

This secondary applies to vertical measurements. Specify a number (*n*) and one of the following units of measurement:

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch

linespace option

LINESP

This establishes a default that can only be used after the **SPACED** option in **DRAWBOX WITHTEXT**, **DRAWCIRCLE WITHTEXT**, and in **SETTEXT**.

linespace value

Specify a number (*n*) and one of the following units of measurement:

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch
POINTS	Points

subcommand word

CORNERLENGTH

length value

Specify the length of the rounded corners for boxes and default length of rounded connections for paths. For boxes and right-angle path connections, corner length is the same as the radius of the corner or connection. Corner lengths are explained in “Corner Length Values with **SETUNITS**” on page 141. Choose from:

SMALL	
MEDIUM	(Default)
LARGE	
MAX	
HALF	

n

IN	Inches
MM	Millimeters
PELS	Pels (default)
BW	Border weight

subcommand word

TEXTMARGIN

text margin

Enter the type of text margin to be used for text placement in boxes and circles. Choose from:

SQUARE	
ROUNDED	(Default)

subcommand word

POSITIONING

positioning option

Choose from:
TOPLEFT (Default)
CENTER

end marker Always end a command with an end marker (;).

The **SETUNITS** command in “Using Default Options (**SETUNITS**)” on page 137 sets up $\frac{1}{4}$ -inch horizontal and vertical defaults and a default line spacing of 0.15 inches.

Sometimes you may want to override **SETUNITS**. For example, a default of $\frac{1}{4}$ inch (shown in “Using Default Options (**SETUNITS**)” on page 137) is not helpful when you want to specify a 0.3-inch measurement. To use the default, you have to determine what 0.3 inches is in terms of 0.25-inch increments (the value is 1.2). It is usually simpler to override the **SETUNITS** command and specify the measurement in inches (.3 IN).

Line Spacing with SETUNITS

SETUNITS can provide default line spacing in two ways:

- You can explicitly define the line spacing with the spacing option, **LINESP**. This default applies to both horizontal and vertical text.
- If **LINESP** is not specified, the default unit of measurement for line spacing is either the primary or the secondary default:
 - If you specify only a primary default, it applies to both horizontal and vertical text.
 - If you specify primary and secondary defaults, the primary default applies to line spacing for text strings that are vertical relative to the overlay. The secondary default applies to line spacing for text strings that are horizontal relative to the overlay. Sample A in Figure 128 uses the primary default as a line spacing unit of measurement. Sample B uses the secondary default.

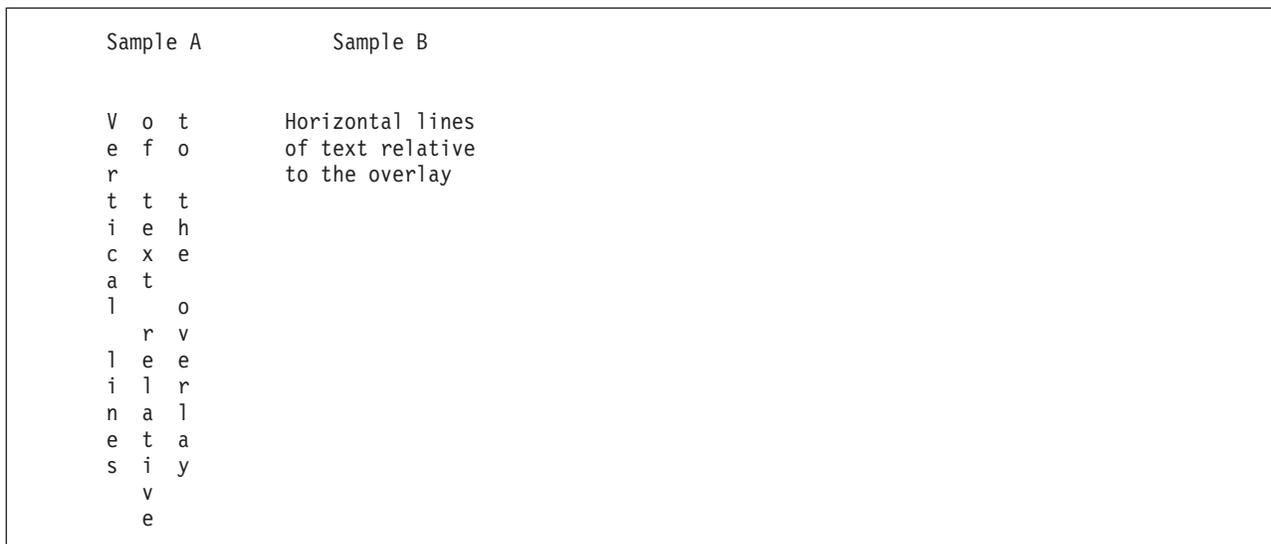


Figure 128. Choosing the Line-Spacing Default

Corner Length Values with SETUNITS

Each rounded box corner and rounded path connection drawn by OGL/370 is an arc of a circle (the *corner arc* or *connection arc*). The degree of rounding - the corner length - is defined by the **CORNERLENGTH** subcommand of **SETUNITS**. It sets the corner length of rounded corners for boxes and the default corner length of rounded connections for paths.

Using Default Options

For a box, the corner length is the same as the radius of the corner arc.

Figure 129 illustrates the corner length of a rounded

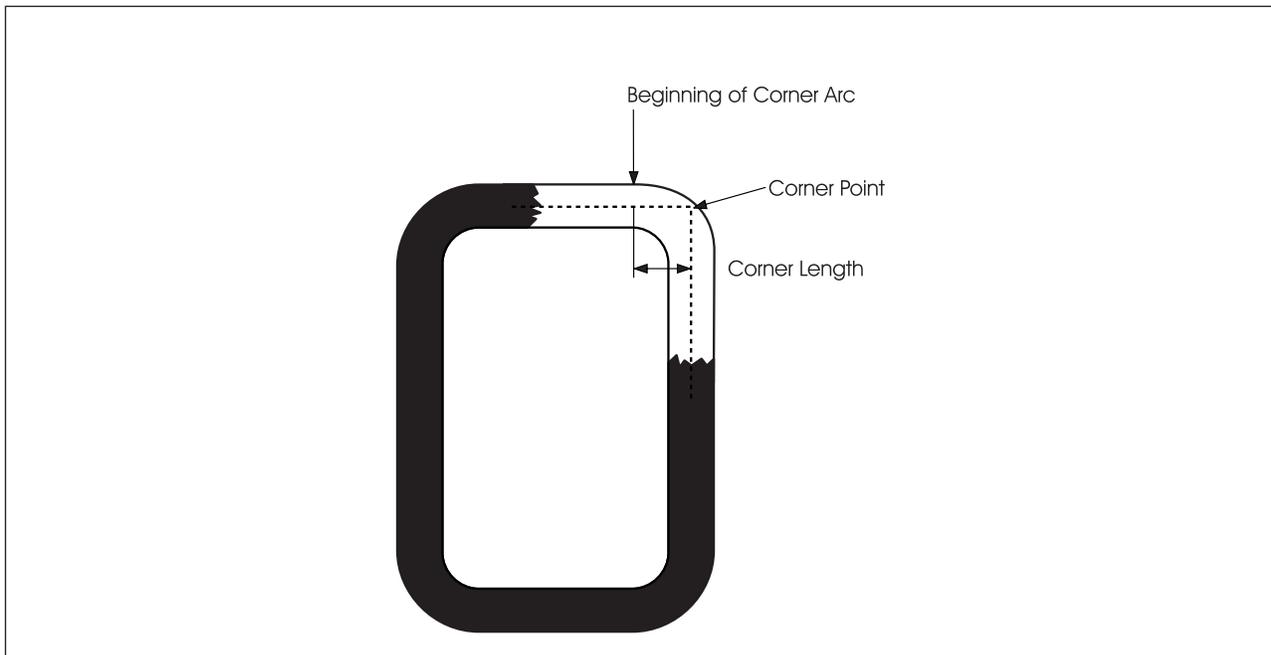


Figure 129. Length of a Rounded Box Corner

For a path, the corner length of a rounded connection is the distance between the corner point and the point on the straight line where the corner arc begins. For a right-angle connection, the corner length is the same as the radius of the connection arc.

Figure 130 illustrates how you determine the exact corner length of a rounded connection.

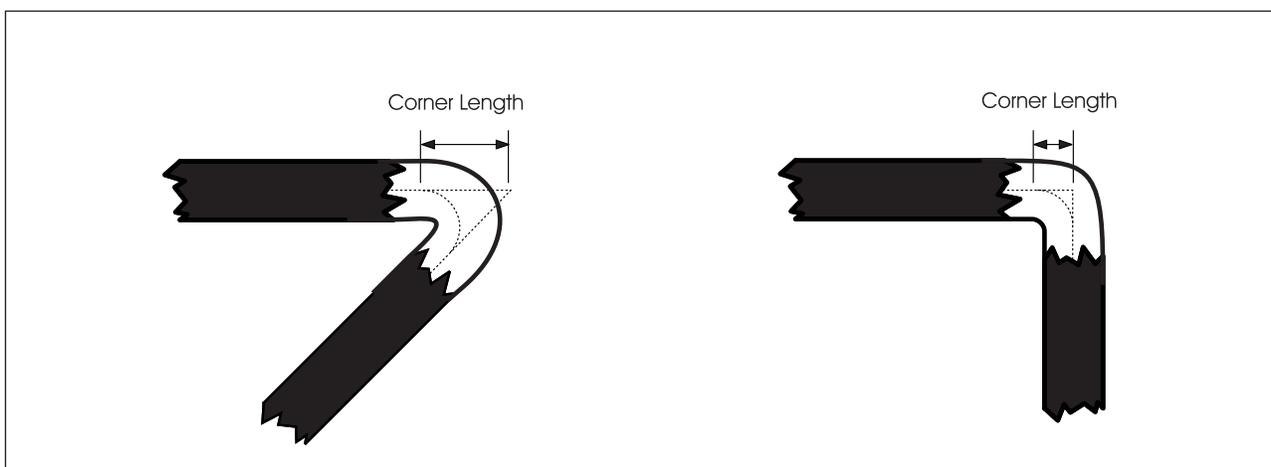


Figure 130. Corner Length of a Rounded Path Connection

You can specify a number (n) and a unit of measurement for corner length or use one of five keywords. You can also define the length as a number of border thicknesses or border weights of the box or path. Choose from:

SMALL The length used is 10 pels.

<u>MEDIUM</u>	The length used is 20 pels (default).
<u>LARGE</u>	The length used is 30 pels.
MAX	For a box, MAX gives a rounded corner with half the length of the shortest side of the box. For a path, MAX gives an arc that extends the full length of the shorter of the two segments being connected.
HALF	For a box, HALF gives a rounded corner with half the length of the shortest side of the box (the same as MAX). For a path, HALF gives an arc that extends half the length of the shorter of the two segments being connected.

Note: See Figure 132 on page 144 and Figure 133 on page 144 for illustrations of **MAX** and **HALF** definitions.

n

IN	Inches
MM	Millimeters
<u>PELS</u>	Pels (default)
BW	Border weight

Note: Using one of the length keywords should satisfy most requirements. Experiment with different lengths and decide which one best suits your overlay.

If you draw a box or path before you set the default corner length with **SETUNITS**, **MEDIUM** is used as the default. You can also set a default for all connections of a given path using the **CONNECTION** subcommand of **DRAWPATH**. This overrides the default set by **SETUNITS**. You can define the corner length of a single connection using the **CONNECTION** option of the TO subcommand of **DRAWPATH**. When you define the corner length for a single connection, this overrides the values set by **SETUNITS** and any value set for the path.

Figure 131 illustrates **SMALL**, **MEDIUM**, and **LARGE** corner lengths for boxes with **MEDIUM** border weight.

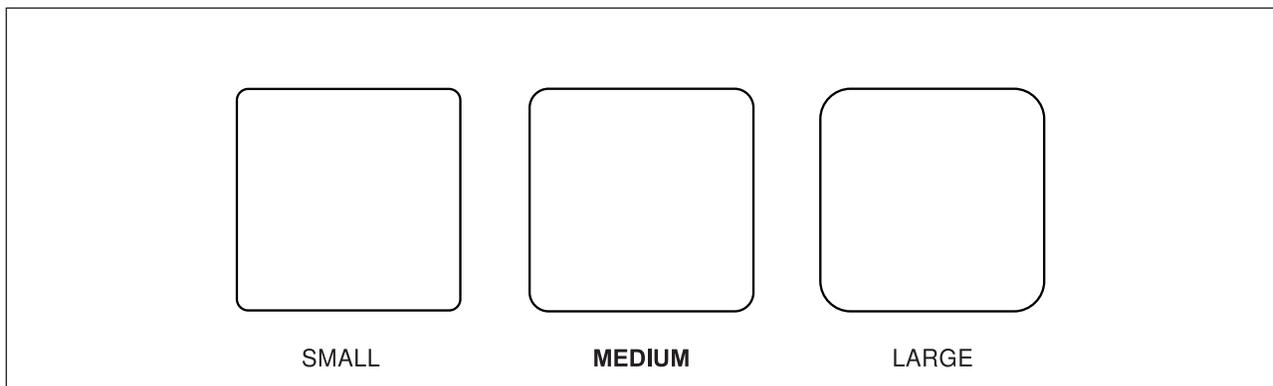


Figure 131. Boxes with **SMALL**, **MEDIUM**, and **LARGE** Corner Length

Figure 132 on page 144 illustrates **HALF** and **MAX** corner lengths for boxes.

Using Default Options

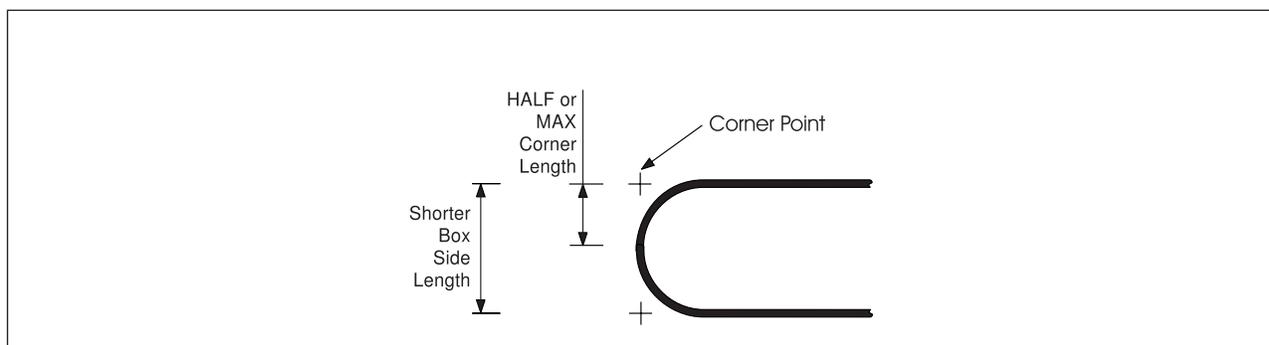


Figure 132. **HALF** and **MAX** Corner Length for Boxes

Figure 133 illustrates **HALF** and **MAX** corner lengths for paths.

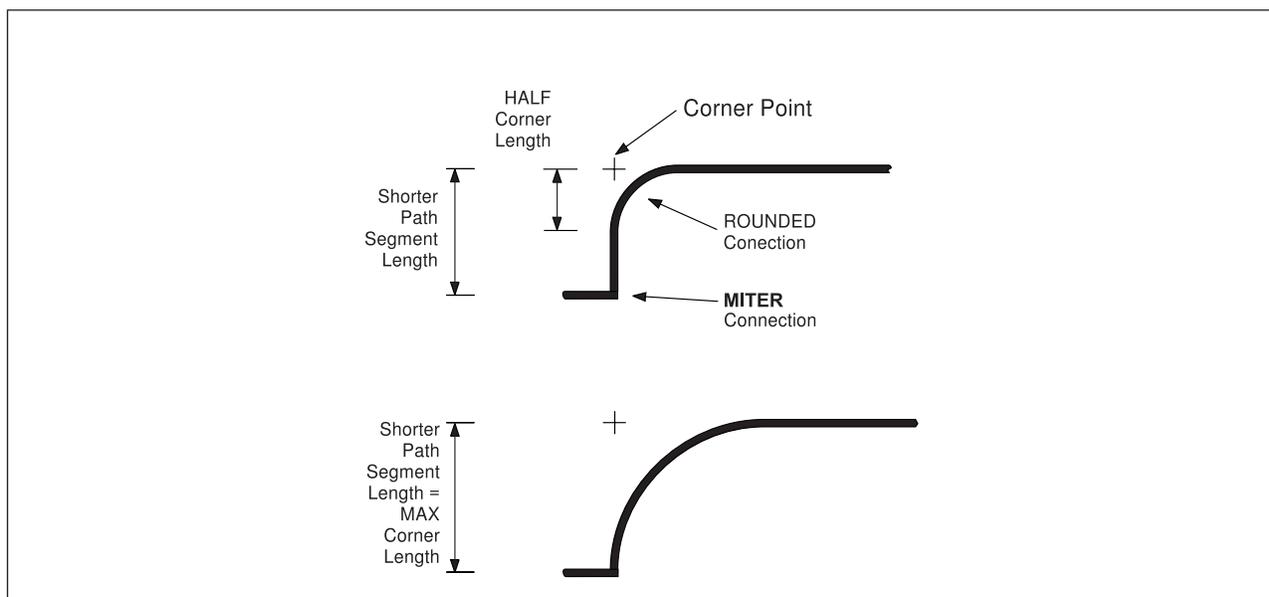


Figure 133. **HALF** and **MAX** Corner Length for Paths

Text Margins with SETUNITS

When you specify text inside a box or circle, the text is placed inside a particular area of the box or circle. This area is called the *text margin*. Text margins are explained in “Text Margins in Boxes” on page 70 and “Text Margins in Circles” on page 73.

Top-Left and Center Positioning with SETUNITS

Suppose that you want OGL/370 to draw a box for you, and you want that box to be positioned at a certain point on the page. There are two ways of specifying the position of a box:

TOPLEFT (Default)
CENTER

These different ways of positioning a box are illustrated in Figure 134 on page 145. In each case, the top-left corner of the box is shown in outline. Notice that the left and right boxes are not in exactly the same place. The box on the left has been positioned with **TOPLEFT** positioning. The top left-hand corner of the box’s border is positioned at the “X”. The box on the right has been positioned with **CENTER** positioning. Notice that the center of the box’s border has been placed at the “X”. Using **CENTER** positioning for boxes is useful if you want to place them close to paths on your overlay, as paths are positioned using a similar method.

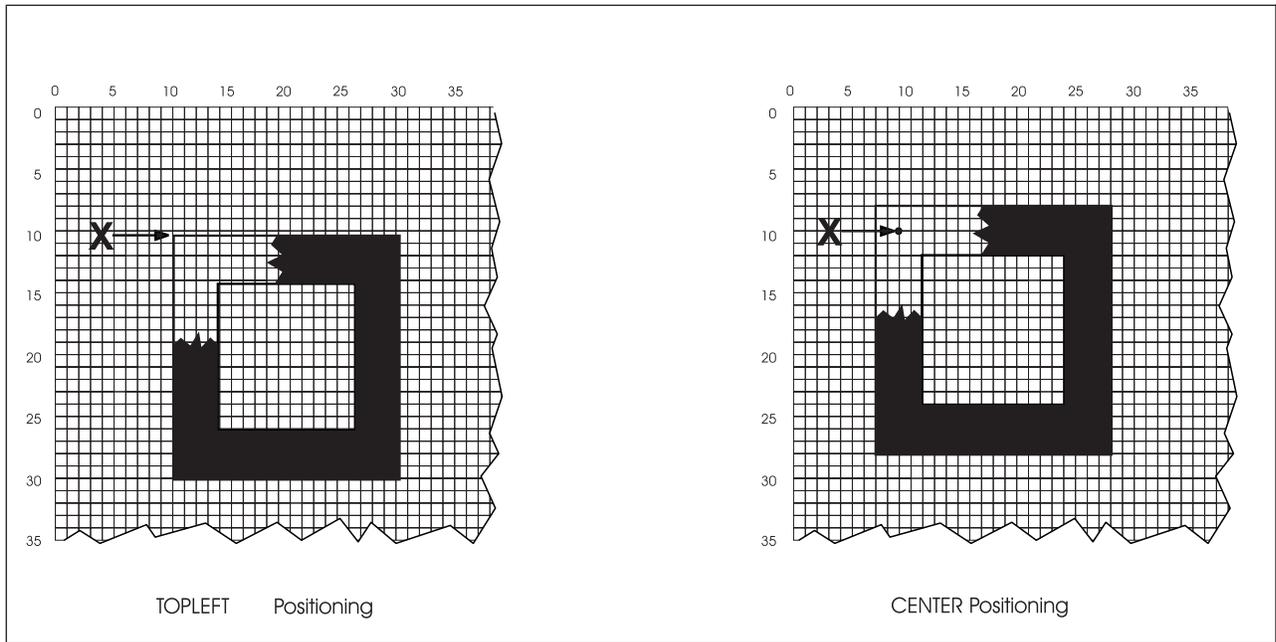


Figure 134. TOPLEFT and CENTER Positioning for a Box

Using Default Options

The positioning option applies to rules in the same way as boxes. See Figure 135.

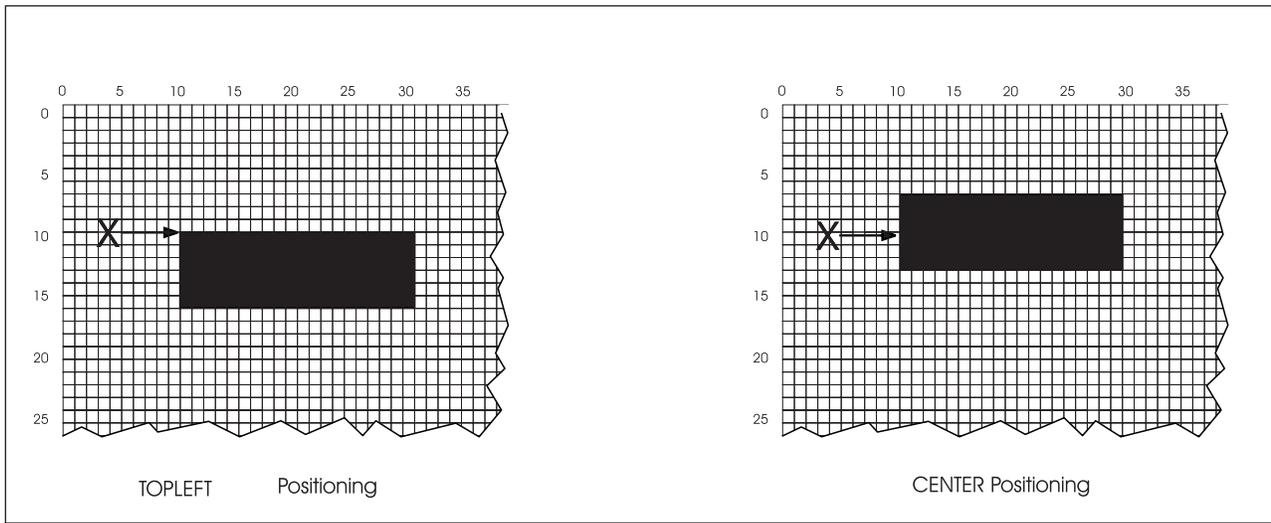


Figure 135. **TOPLEFT** and **CENTER** Positioning for a Rule

Figure 136 on page 147 is used throughout the rest of this chapter for demonstration purposes. This is the top portion of the overlay shown in Figure 1 on page 3. In this chapter, we treat the top portion as a separate overlay.

Although this sample is more complex than the one from the previous chapter, by using the additional features explained in this chapter, your overlay definition is not much longer than that of the last overlay definition.

Using Default Options

```
DRAWMASK .25 IN;
```

- Use the **SETUNITS** default as the unit of measurement and explicitly state the number of units to be used in the mask. If **SETUNITS** establishes $\frac{1}{4}$ -inch as the horizontal and the vertical default, the mask can then be specified as:
-

```
DRAWMASK 1 1;
```

or

```
DRAWMASK 1;
```

- Use the **SETUNITS** default as the unit of measurement and default in **DRAWMASK** to 1 unit. The $\frac{1}{4}$ -inch mask is specified as:
-

```
DRAWMASK;
```

Here are the commands to get you started with the overlay definition for sample PREREG:

```
- 'GETTING STARTED'  
  SETUNITS .25 IN .25 IN LINESP .15 IN;  
  OVERLAY PREREG SIZE 29 25 OFFSET 0 2;  
  ORIENT 0;  
  CONTROL NOSTORE ALL;  
  DRAWMASK;
```

Send this definition to the printer to get a mask the same size as the form. The mask is useful in understanding the remaining material.

Repeating Rules (DRAWRULE REPEAT)

In Figure 137 the rules that are important are numbered. Note that rules 1 through 10 are identical, as are rules 11 through 14. In the last chapter, we defined each rule separately. In this chapter, learn how you to define identical rules in a single **DRAWRULE** command with a **REPEAT** subcommand.

PRE-REGISTRATION

Please review the information on this form. During your pre-registration interview, tell your advisor of any required corrections. Information on this form is strictly confidential and will not be released without your consent.

TropiCal Community College

1 2 3 4 5 6 7 8 9 10

11 12 13 14

LAST NAME FIRST NAME MI

SOCIAL SECURITY NUMBER

STREET ADDRESS OR BOX NUMBER

CITY STATE

ZIP CODE

HOME PHONE

WORK PHONE

Figure 137. Repeated Rules

For the **REPEAT** subcommand, the rules that are to be repeated must be identical in direction, length, thickness, and type.

There are two ways to repeat rules:

Spaced Repetition: Enter an equal distance between rules.

Location Repetition: Enter the location of each rule.

If you answer “yes” to both of the following questions, use spaced repetition; otherwise, you must use location repetition.

1. Are the rule origins lined up either vertically or horizontally?
2. Is the spacing between all the rules equal?

Repeating Rules

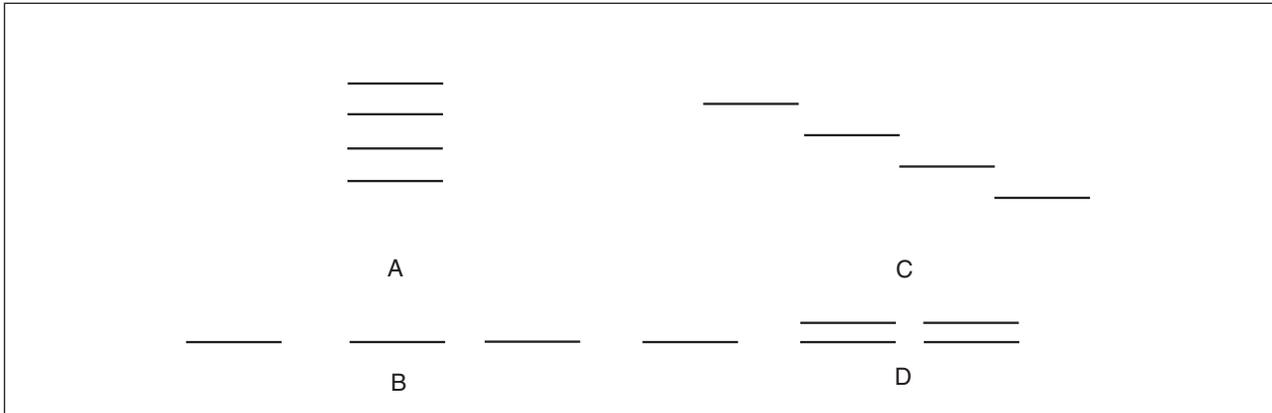


Figure 138. Spaced and Location Rule Repetition

In Figure 138:

- The rule origins in group A are lined up vertically, and those in group B are lined up horizontally. However, in C and D, the origins are not lined up either vertically or horizontally.
- The spacing is equal for A and C, but unequal for B and D.

You can answer “yes” to both questions only for group A. Therefore, only group A can be defined using spaced repetition. Examples B, C, and D must be defined using location repetition.

The **REPEAT** subcommand follows the rule type entry. Spaced and location repetitions are explained separately.

Spaced Repetition: The **REPEAT** subcommand has the following entries:

DRAWRULE	DOWN	1	<u>MEDIUM</u>	<u>SOLID</u>	
command word	direction	length	thickness	rule type	
REPEAT	<u>ACROSS</u>	9	SPACED	1	;
subcommand word	direction	repetitions	spacing word	spacing value	end marker

Figure 139. **DRAWRULE REPEAT** (Spaced Repetition)

subcommand word

REPEAT

rule direction This entry refers to the direction in which the rules are repeated. The direction of the rule is specified in the first part of the command. Choose from:

ACROSS (Default)
DOWN

repetitions The number (*n*) of additional rules.

spacing word **SPACED**

spacing value A number (*n*) and unit of measurement that specify the horizontal distance (after **ACROSS**) or the vertical distance (after **DOWN**) from one rule to the next. The unit of measurement may either default to the **SETUNITS** unit or may be written as one of the following:

n

IN Inches
MM Millimeters
PELS Pels

Note: Rule thickness is a factor that you must consider when you are specifying the spacing value. Figure 140 illustrates some examples.

end marker Always end a command with an end marker (;).

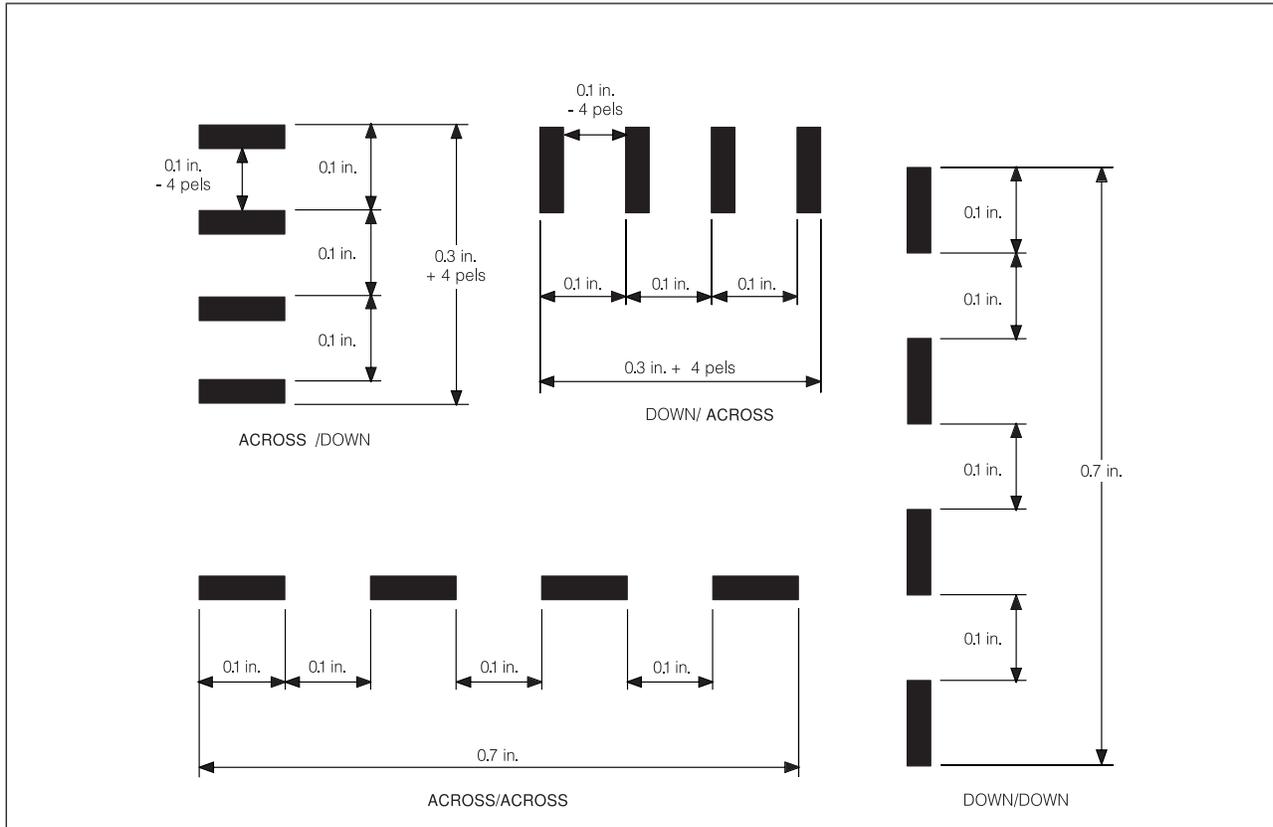


Figure 140. Spaced Repetition for **DRAWRULE**. (Not actual size.)

ACROSS/DOWN

This part of Figure 140 illustrates a rule, drawn in the direction, and repeated three times in the **DOWN** direction with a spacing of 0.1 inches.

DOWN/ACROSS

This part of Figure 140 illustrates a rule, drawn in the **DOWN** direction, and repeated three times in the **ACROSS** direction with a spacing of 0.1 inches.

ACROSS/ACROSS

This part of Figure 140 illustrates a rule, drawn in the **ACROSS** direction, and repeated three times in the **ACROSS** direction with a spacing of 0.1 inches.

DOWN/DOWN This part of Figure 140 illustrates a rule, drawn in the **DOWN** direction, and repeated three times in the **DOWN** direction with a spacing of 0.1 inches.

In the **ACROSS/DOWN** and **DOWN/ACROSS** examples, the distance between the rules is 0.1 inches minus the thickness of 1 rule (4 pels). In the **ACROSS/DOWN** example, the distance from the top of the first rule to the bottom of the last rule is 0.4 inches plus the thickness of 1 rule.

Repeating Rules

For most overlay designs, a 4-pel difference is unimportant. However, if between 2 of the rules in the **DOWN/ACROSS** example, you print a character that is just under 0.1 inches wide, the result is unsatisfactory. To get exactly 0.1 inches between the rules, you have to specify the spacing as 0.1 inches plus 4 pels, or 28 pels.

Figure 139 on page 150 shows the **DRAWRULE** command for rules 1 through 10. We have assumed a 0.25-inches **SETUNITS** default value; and the command is preceded by the following **POSITION** command, which defines the coordinates for rule 1:

```
POSITION ABSOLUTE 17 ABSOLUTE 5;
```

Location Repetition: The **REPEAT** subcommand has the following entries:

subcommand word

REPEAT

location option **LOCATION**

This word identifies the type of repetition and indicates that what follows are the coordinates for the rule origin. Enter this word and the coordinates for *each* repetition of the rule.

horizontal coordinate

A number (*n*) and unit of measurement that specify the horizontal coordinate of the origin of the rule measured from the overlay origin.⁸ The unit of measurement can default to the **SETUNITS** value, or you can specify one of the following:

n

IN	Inches
MM	Millimeters
PELS	Pels

vertical coordinate

A number (*n*) and unit of measurement that specify the vertical coordinate of the rule origin. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

end marker Always end a command with an end marker (;).

Figure 141 on page 153 shows the **DRAWRULE REPEAT** command for rules 11 through 14 (shown in Figure 137 on page 149). We have assumed a 0.25-inches **SETUNITS** default value; and the command is preceded by the following **POSITION** command, which defines the coordinates for rule 11:

```
POSITION ABSOLUTE 19 ABSOLUTE 8;
```

8. If the command is part of a **GROUP** definition (described in “Defining and Placing Groups” on page 162), the coordinates are measured from the group origin.

DRAWRULE	<u>ACROSS</u>	5	<u>MEDIUM</u>	<u>SOLID</u>
command word	direction	length	thickness	rule type
REPEAT	LOCATION	1	10	
	LOCATION	16	10	
	LOCATION	22	10	;
subcmd word	location option	horizontal coordinate	vertical coordinate	end marker

Figure 141. DRAWRULE REPEAT

Repeating Boxes

Repeating Boxes (DRAWBOX REPEAT)

You can repeat boxes in much the same way as you repeat rules. However, where the repeated rules had to be completely identical, this is not so with boxes. Although repeated boxes must be the same size and have the same border thickness, they can have different shading and text.

In Figure 142 on page 155, the boxes that are important are numbered.

Notes:

1. Boxes 1 through 4 are identical except for text.
2. Boxes 5 through 7 are completely identical.
3. Boxes 8 through 12 are identical except for shading.

As with rules, there are two ways to repeat boxes:

Spaced Repetition: Enter an equal distance between boxes.

Location Repetition: Enter the location of each box.

If you answer “yes” to both of the following questions, use spaced repetition; otherwise, you must use location repetition:

1. Are the box origins lined up either vertically or horizontally?
2. Is the spacing between all the boxes equal?

In Figure 143 on page 155:

- The box origins in group A are lined up vertically, and those in group B are lined up horizontally. However, in C and D, the origins are not lined up either horizontally or vertically.
- The spacing is equal for A and C but unequal for B and D.

You can answer “yes” to both questions only for group A. Therefore, only group A can be defined using spaced repetition. Examples B, C, and D must be defined using location repetition.

Repeating Boxes

1

STREET ADDRESS OR BOX NUMBER

CITY

STATE

2

ZIP CODE

3

HOME PHONE

4

WORK PHONE

199__

SE
M
E
S
T
E
R

CURRENT SEMESTER ENROLLMENT										
Course and section No.	Cr Hrs	Course Title	Room	Time	M	T	W	T	F	

LEGAL RESIDENCE

5 Tropi County

6 California

7 USA

199__

SE
M
E
S
T
E
R

TO BE COMPLETED DURING ADVISING										
Course and section No.	Cr Hrs	Course Title	Room	Time	M	T	W	T	F	

Routing Code (office use only)

8-9-10-11-12

Figure 142. Repeated Boxes

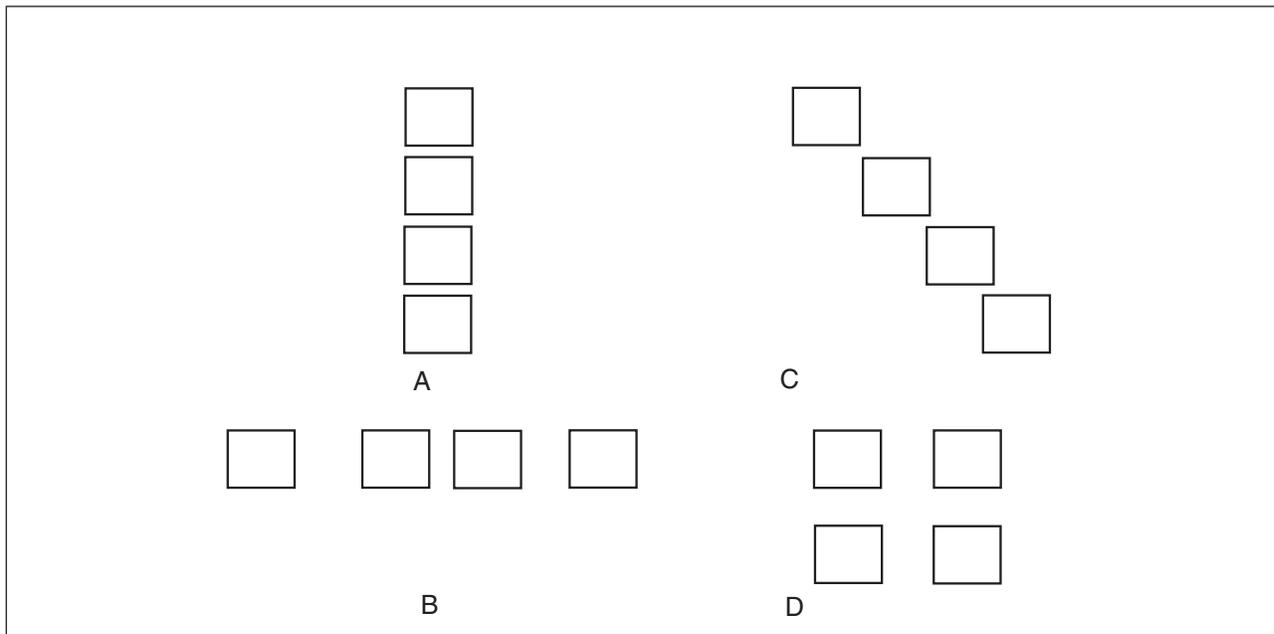


Figure 143. Spaced and Location Box Repetition

The **REPEAT** subcommand follows the **DIAGONAL** subcommand, if it has been specified.

Repeating Boxes

Spaced Repetition: The **REPEAT** subcommand has the following entries:

DRAWBOX	.2 in	.2 in	<u>MEDIUM</u>	<u>SOLID</u>
command word	box width	box height	border thickness	border type
REPEAT	DOWN	2	SPACED	.2 in ;
subcmd word	direction	repetitions	spacing option	spacing value end marker

Figure 144. **DRAWBOX REPEAT**

subcommand word

REPEAT

direction This word refers to the direction in which the boxes are repeated. Choose from:

ACROSS (Default)
DOWN

repetitions The number (*n*) of additional boxes.

spacing word **SPACED**

spacing value A number (*n*) and a unit of measurement that specify the horizontal distance (after **ACROSS**) or the vertical distance (after **DOWN**) between repeated boxes. The unit of measurement can default to the **SETUNITS** value, or you can specify one of the following:
n

IN Inches
MM Millimeters
PELS Pels

end marker Always end a command with an end marker (;).

In Chapter 2, “Designing a Simple Overlay” on page 23, you learned that a box is measured from the outside of the left border to the inside of the right border, and from the outside of the top border to the inside of the bottom border.

For boxes repeated **ACROSS**, the spacing between boxes is measured from the inside of the right border of one box to the outside of the left border of the next box. For boxes repeated **DOWN**, the spacing is measured from the inside of the bottom border of one box to the outside of the top border of the next box.

If you specify a 0.25-inches by 0.25-inches box with a **MEDIUM** border to be repeated **ACROSS** three times at 0.2-inch intervals, this is what you get:

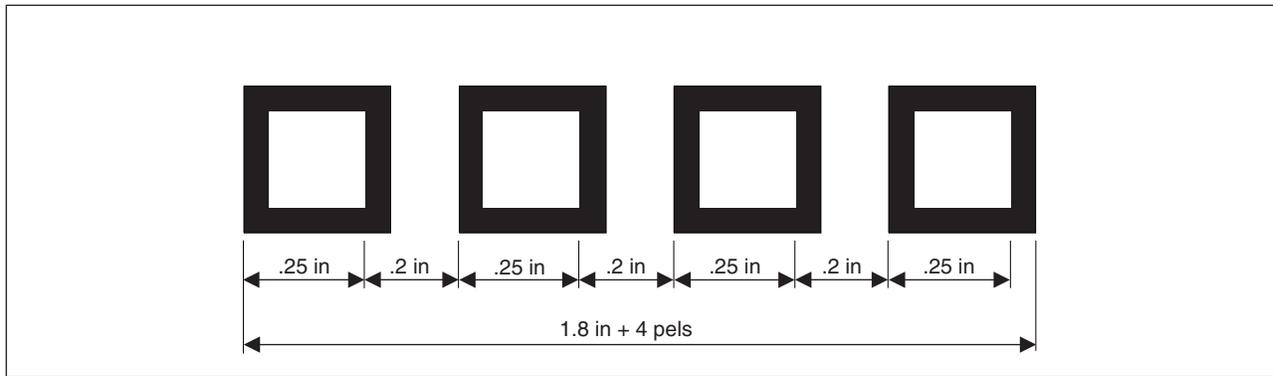


Figure 145. Spaced Boxes. (Not actual size.)

Note: In Figure 145 the total width of the boxes and the spaces between the boxes is 1.8 inches plus the thickness of one box border, in this case is 4 pels. For most overlays, the box border is not important.

Setting the spacing value to 0 produces boxes that touch each other, as do the “Routing Code” boxes (8 through 12) in Figure 142 on page 155. For example, a 0.25-inches by 0.25-inches box with a bold border repeated four times at 0 intervals, looks like the illustration in Figure 146.

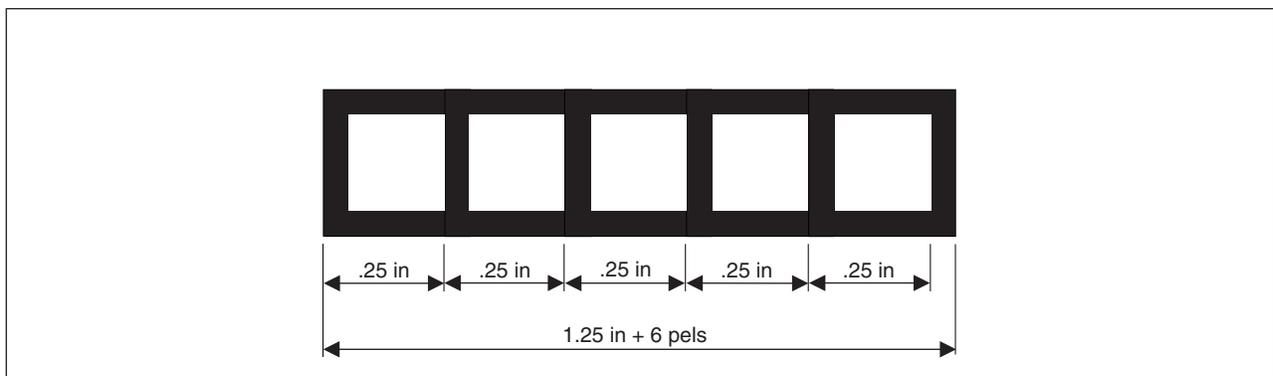


Figure 146. Boxes Spaced 0. (Not actual size.)

Note: The thickness of one box border is part of the total width and height of boxes.

Figure 144 on page 156 shows the **DRAWBOX** command for boxes 5 through 7 (shown in Figure 142 on page 155.). The **POSITION** command positions the first box relative to the origin of the horizontal rule in the box entitled “LEGAL RESIDENCE”.

```
POSITION RIGHT .05 IN DOWN .3 IN;
```

Write the **DRAWBOX** command for boxes 8 through 12, assume a 0.25 **SETUNITS** default. Do not write the shading commands now. Compare your version to the following figure.

```
POSITION ABSOLUTE 23 ABSOLUTE 22.5;
POSITION UP 6 PELS LEFT 15 PELS;
DRAWBOX 1 1 10 SOLID
      REPEAT ACROSS 4 SPACED 0;
```

Repeating Boxes

Chances are, you did not have an absolute and a relative **POSITION** command. In our sample overlay, the “routing code” boxes, with their thick borders, would have exceeded the overlay frame if we had not moved them slightly to the left.

Location Repetition: The **REPEAT** subcommand has the following entries:

DRAWBOX	5	1.5	MEDIUM	SOLID
command word	box width	box height	border thickness	border type
REPEAT	LOCATION	1	9	
	LOCATION	16	9	
	LOCATION	22	9	;
subcmd word	location option	horizontal coordinate	vertical coordinate	end marker

Figure 147. **DRAWBOX REPEAT** (Location Repetition)

subcommand word

REPEAT

location option **LOCATION**

This word identifies the type of repetition and indicates coordinates for the box origin that follows. You must enter this word and the coordinates for each repetition of the box.

horizontal coordinate

A number (*n*) and unit of measurement that specify the horizontal coordinate of the origin of the box measured from overlay origin.⁸ The unit of measurement can default to the **SETUNITS** unit, or you can specify one of the following:

n

IN Inches
MM Millimeters
PELS Pels

vertical coordinate

A number (*n*) and unit of measurement that specify the vertical coordinate of the origin of the box measured from the overlay origin.⁸ The unit of measurement can default to the **SETUNITS** value, or you can specify one of the following:

n

IN Inches
MM Millimeters
PELS Pels

Figure 147 shows the **DRAWBOX REPEAT** command for boxes 1 through 4 in Figure 142 on page 155. We have assumed a 0.25-inches **SETUNITS** default value, and the command is preceded by the following **POSITION** command, which defines the coordinates for box 1:

```
POSITION ABSOLUTE 19 ABSOLUTE 7;
```

Note: Text has not been added to the boxes yet.

Varying Shading and Text

You can specify the shading of the individual boxes with a **REPEAT** subcommand, and you can put different text in different boxes. To do either of these things, you must know the number of the box or boxes to which the shading or text applies. Determining the box number depends on how the boxes are repeated.

Location Repetition: The boxes are numbered beginning with 1 in the order in which you defined them. For example, in Figure 148, the boxes are numbered 1 to 4 according to the order in which their coordinates appear in the following command.

```

DRAWBOX 5 1.5 MEDIUM SOLID      -'DEFINES BOX 1'
      REPEAT LOCATION 1 9        -'DEFINES BOX 2'
            LOCATION 16 9        -'DEFINES BOX 3'
            LOCATION 22 9;       -'DEFINES BOX 4'

```

Information on this form is strictly confidential and will not be released without your consent.

CTC

LAST NAME	FIRST NAME	MI	SOCIAL SECURITY NUMBER			
STREET ADDRESS OR BOX NUMBER			1	CITY		STATE
2	ZIP CODE		3	HOME PHONE		4
WESTER E...			...		LEG	

Figure 148. Numbering Boxes, Created with Location Repetition

Spaced Repetition: The boxes are numbered beginning with 1 from left to right (for **ACROSS**) or from top to bottom (for **DOWN**). For example, in Figure 149 on page 160 the boxes are numbered as OGL/370 sees them.

Repeating Boxes

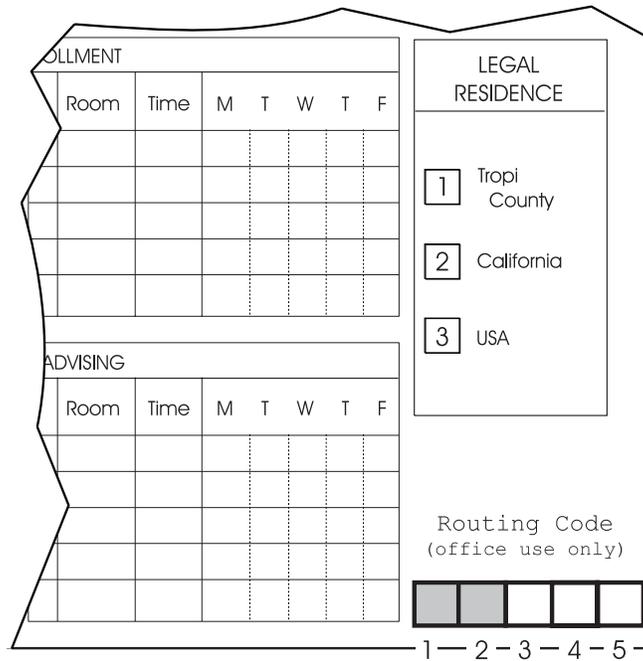


Figure 149. Numbering Boxes, Created with Spaced Repetition

Varying Shading

After you have written the entire **REPEAT** subcommand, you can specify shade.

The shading option has the following parts:

DRAWBOX	1	1	10	SOLID						
command word	box width	box height	border thickness	border type						
REPEAT	ACROSS	4	SPACED	0	SHADE	BOX 1	STANDARD	LIGHT		
subcommand word	direction	repetitions	spacing option	spacing value	shading option	box	shade pattern	shade type	end marker	
						BOX 2	STANDARD	LIGHT	;	

Figure 150. DRAWBOX REPEAT (Shading)

shading option **SHADE**

box Specify the box or boxes to be shaded. Choose from:
ALL All the boxes have the same shading (default).
BOX n All the boxes do not have the same shading. For each box to be shaded, enter **BOX** followed by the box number and the shade pattern and type.

shade pattern and shade type

Shade patterns and shade types are the same as those described in "Shading" on page 44.

The first two routing boxes in overlay PREREG have **LIGHT** shading.

Varying Text

If all the boxes have the same text, you specify the text with one **WITHTEXT** subcommand. Otherwise, the text for each box must be specified in a separate **WITHTEXT** subcommand. You can include one or more **WITHTEXT** subcommands within a single **DRAWBOX** command.

The **WITHTEXT** subcommands appear after the **SHADE** subcommand, or if there is no **SHADE** subcommand, after the border type specification.

The only change to the subcommand presented in Chapter 2, “Designing a Simple Overlay” on page 23 is the addition of the following entry immediately after the subcommand word **WITHTEXT**:

DRAWBOX	5	1.5	<u>MEDIUM</u>	<u>SOLID</u>					
command word	box width	box height	border thickness	border type					
REPEAT	LOCATION	1		9					
	LOCATION	16		9					
	LOCATION	22		9					
subcommand word	location option	horizontal coordinate	vertical coordinate						
WITHTEXT	BOX 1	<u>0</u>	<u>MODERN</u>	BOTTOM	<u>CENTER</u>	LINE	font5	'CITY'	
WITHTEXT	BOX 2	<u>0</u>	<u>MODERN</u>	BOTTOM	<u>CENTER</u>	LINE	font5	'ZIP CODE'	
WITHTEXT	BOX 3	<u>0</u>	<u>MODERN</u>	BOTTOM	<u>CENTER</u>	LINE	font5	'HOME PHONE'	
WITHTEXT	BOX 4	<u>0</u>	<u>MODERN</u>	BOTTOM	<u>CENTER</u>	LINE	font5	'WORK PHONE' ;	
subcmd word	box	orientation	format	placement	subcmd word	font	text	end marker	

Figure 151. **DRAWBOX REPEAT (WITHTEXT)**

box Specify the box or boxes in which text is to be included. Choose from:

ALL All the boxes have the same text.

BOX *n* All the boxes do not have the same text. In each **WITHTEXT** subcommand, enter **BOX** followed by the appropriate number.

BOX 1 (Default)

Figure 148 on page 159 illustrates different text in boxes 1 through 4.

Defining and Placing Groups

Defining and Placing Groups

Except for the text “CURRENT SEMESTER ENROLLMENT”, the overlay section shown in Figure 152 occurs elsewhere on the overlay. You can repeat this section by rewriting the original commands with different values in the **POSITION** commands.

However, OGL/370 has a feature that saves you the time of figuring out and entering all the changes. The commands that define an overlay section are grouped between two other commands (**DEFINE** and **ENDDF**), and the entire group is given a name. This group can then be positioned and placed as often as you wish.

S E M E S T E R	199	CURRENT SEMESTER ENROLLMENT									
		Course and section No.	Cr Hrs	Course Title	Room	Time	M	T	W	T	F
	FALL										
	SPR										
	SUM										

Figure 152. A Group

Positioning the group and items within the group are explained in the following sections.

Only the following commands can be specified in a group:

DRAWBOX
DRAWCIRCLE
DRAWGRAPHIC
DRAWMASK
DRAWPATH
DRAWRULE
POSITION
SETTEXT
SETUNITS

This restriction means that you cannot include an image pattern or a segment in a group. For example, the section in the upper-right corner of the sample overlay PREREG consists of text, an image pattern, and a page segment. Only the text can be part of a group.

Another restriction to remember is that a group cannot be rotated. Therefore, if you want a section of an overlay printed in two different orientations on the overlay, you must define the section twice.

Positioning a Group (POSITION)

A group, like a box or an image, has an origin. The group origin is the point from which all the items in the group are measured. It helps to imagine the group as a rectangle and the origin of the group as the upper-left corner of the rectangle. When you want to put the group on the overlay, you position the group in terms of its origin. The **POSITION** command for the first group in the sample overlay is:

```
POSITION ABSOLUTE 0 ABSOLUTE 11;
```

The **POSITION** command for the second group is:

```
POSITION ABSOLUTE 0 ABSOLUTE 17.5;
```

Defining a Group (DEFINE GROUP)

Defining a group is simple, even if the commands contained in the group are complex. The **DEFINE** command has the following parts:

```

DEFINE  sched      GROUP      ;
command section  section  end
word    name     type     marker

group command ;
.
.
group command ;

group      end
command(s) marker(s)

ENDDDEF   ;

command end
word     marker
    
```

Figure 153. **DEFINE GROUP** Commands. The **DEFINE** and **ENDDDEF** commands and each of the group commands must begin on a new line.

command word

DEFINE

section name The name used when you place the group. The section name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be more longer than eight characters.

section type **GROUP**

End the **DEFINE** command with a semicolon. When OGL/370 receives this command, it is notified that the following commands are part of a group.

group commands

Write the allowable commands (listed on the previous page) in the usual way. Each command must, of course, be ended with a semicolon. The end of the entire group is signaled with the simple command **ENDDDEF**, which is the next entry.

command word

ENDDDEF

end marker Always end a command with an end marker (;).

Defining and Placing Groups

The **DEFINE GROUP** commands in Figure 153 on page 163 are the commands used for the class-schedule group, Figure 152 on page 162, named “SCHEM”. You are almost ready to write the definition of a group. However, first you must understand how the **POSITION** and **SETUNITS** commands work in relation to a group.

Positioning inside a Group (**POSITION**)

The various items in a group can be positioned either in relation to the group origin or relative to the previous **POSITION** command within the group.

The absolute **POSITION** command, when written inside of a group definition, is measured from the group origin, not from the overlay origin.

The relative **POSITION** command, when written inside of a group definition, is measured from the last **POSITION** command, whether inside of or outside of the group definition. However, any **POSITION** command inside of a group is ignored when OGL/370 uses a relative **POSITION** command to position an item outside of the group.

Using **SETUNITS** inside a Group (**SETUNITS**)

SETUNITS operates much the same way that **POSITION** does: A **SETUNITS** command outside of the group definition is valid inside of the group unless overridden by a **SETUNITS** command inside of the group. But a **SETUNITS** command inside of a group definition has no effect on the commands that follow it outside of the group.

Placing a Group (**PLACE**)

The **PLACE** command is used for a group in the same way it is used for an image pattern. After you have written the **POSITION** command for the group, immediately follow it by a **PLACE** command that identifies both the type and the name of the item that is being placed.

See “Defining, Positioning, and Placing a Group” for a description of the commands to define, position, and place the schedule group in the sample overlay. Notice that the commands to place the headings:

"CURRENT SEMESTER ENROLLMENT"

and

"TO BE COMPLETED DURING ADVISING"

are outside of the group definition.

Defining, Positioning, and Placing a Group

Note: The following figure includes the **SETTEXT** command, which is described later in this chapter.

```

-'STUDENT SCHEDULE GROUP'
  DEFINE SCHED GROUP;
  SETUNITS .2 IN .2 IN LINESP .1 IN;

-TEXT
  POSITION ABSOLUTE 2.5 ABSOLUTE 1;
  SETTEXT 0 MODERN LEFT
    LINE FONT1 NOUNDERLINE CHAR '198'
      UNDERLINE CHAR ' ';
  POSITION ABSOLUTE 1.5 ABSOLUTE 1;
  SETTEXT 0 COLUMN TOP
    LINE FONT3 NOUNDERLINE CHAR 'SEMESTER';
-'BOXES & TEXT'
  POSITION ABSOLUTE 1.25 IN ABSOLUTE 0;           -'MAIN BOX'
  DRAWBOX 4.4 IN 1.5 IN MEDIUM SOLID;
  POSITION RIGHT .04 IN DOWN .35 IN;
  SETTEXT 0 MODERN CENTER SPACED 1             -'COURSE NO.'
    LINE FONT5 NOUNDERLINE CHAR 'Course and'
    LINE FONT5 NOUNDERLINE CHAR 'Section No.';
  POSITION RIGHT 4 DOWN 0;
  SETTEXT 0 MODERN CENTER SPACED 1             -'CR HRS'
    LINE FONT5 NOUNDERLINE CHAR 'Cr'
    LINE FONT5 NOUNDERLINE CHAR 'Hrs';
  POSITION RIGHT 3 DOWN .05 IN;
  SETTEXT 0 MODERN CENTER                       -'COURSE TITLE'
    LINE FONT5 NOUNDERLINE CHAR 'Course Title';
  POSITION RIGHT 6 DOWN 0;
  SETTEXT 0 MODERN CENTER                       -ROOM
    LINE FONT5 NOUNDERLINE CHAR 'Room';
  POSITION RIGHT 2 DOWN 0;
  SETTEXT 0 MODERN CENTER                       -TIME
    LINE FONT4 NOUNDERLINE CHAR 'Time';
  POSITION ABSOLUTE 4.65 IN ABSOLUTE 1;
  DRAWBOX 1 1.5 0                               -DAYS
    REPEAT ACROSS 4 SPACED 0
      WITHTEXT BOX 1 0 MODERN CENTER CENTER
        LINE FONT5 NOUNDERLINE CHAR 'M'
      WITHTEXT BOX 2 0 MODERN CENTER CENTER
        LINE FONT5 NOUNDERLINE CHAR 'T'
      WITHTEXT BOX 3 0 MODERN CENTER CENTER
        LINE FONT5 NOUNDERLINE CHAR 'W'
      WITHTEXT BOX 4 0 MODERN CENTER CENTER
        LINE FONT5 NOUNDERLINE CHAR 'T'
      WITHTEXT BOX 5 0 MODERN CENTER CENTER
        LINE FONT5 NOUNDERLINE CHAR 'F';

```

Defining and Placing Groups

```
-LINES
  POSITION ABSOLUTE 1.25 IN ABSOLUTE 1;           -HORIZONTAL
  DRAWRULE ACROSS 4.4 IN MEDIUM SOLID;
  POSITION RIGHT 0 DOWN 1.5;
  DRAWRULE ACROSS 4.4 IN MEDIUM SOLID
    REPEAT DOWN 4 SPACED 1;
  POSITION ABSOLUTE 2.05 IN ABSOLUTE 1;           -VERTICAL
  DRAWRULE DOWN 1.3 IN MEDIUM SOLID
    REPEAT LOCATION 2.35 IN 1
      LOCATION 3.85 IN 1
      LOCATION 4.25 IN 1
      LOCATION 4.65 IN 1;
  POSITION ABSOLUTE 4.85 IN ABSOLUTE 2.5;
  DRAWRULE DOWN 5 MEDIUM DASHED
    REPEAT ACROSS 3 SPACED 1;
-'SEMESTER BOXES'
  POSITION ABSOLUTE 2.5 ABSOLUTE 2.5;
  DRAWBOX 2.5 1 MEDIUM
    REPEAT DOWN 2 SPACED 1;
  POSITION RIGHT 1.5 DOWN 0;
  DRAWRULE DOWN 1 MEDIUM SOLID
    REPEAT DOWN 2 SPACED 1;
  POSITION ABSOLUTE 2.75 ABSOLUTE 3.25;
  SETTEXT 0 MODERN LEFT
    LINE FONT5 NOUNDERLINE CHAR 'FALL';
  POSITION RIGHT 0 DOWN 2;
  SETTEXT 0 MODERN LEFT
    LINE FONT5 NOUNDERLINE CHAR 'SPR';
  POSITION RIGHT 0 DOWN 2;
  SETTEXT 0 MODERN LEFT
    LINE FONT5 NOUNDERLINE CHAR 'SUM';
ENDEF;
-'PLACE THE GROUP TWICE'
  POSITION ABSOLUTE 0 ABSOLUTE 11;
  PLACE GROUP SCHED;
  POSITION RIGHT 10.25 DOWN .6;
  SETTEXT 0 MODERN LEFT
    LINE FONT5 NOUNDERLINE CHAR 'CURRENT SEMESTER ENROLLMENT';
  POSITION ABSOLUTE 0 ABSOLUTE 17.5;
  PLACE GROUP SCHED;
  POSITION RIGHT 9.75 DOWN .6;
  SETTEXT 0 MODERN LEFT
    LINE FONT5 NOUNDERLINE CHAR 'TO BE COMPLETED DURING ADVISING';
```

More Features of DRAWBOX

You have already seen how to draw simple boxes using the **DRAWBOX** command. This section describes some additional features of **DRAWBOX**, such as how to define boxes with rounded corners or diagonals. Figure 154 shows a box with some of these features. The commands in this section are for this box.

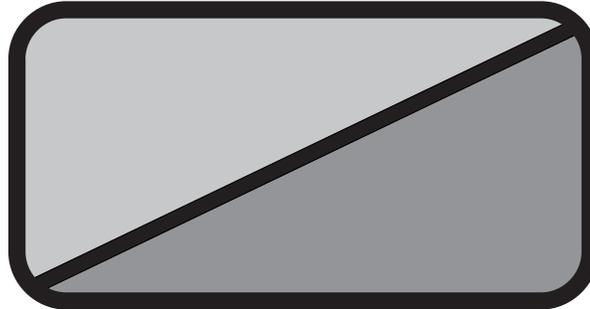


Figure 154. Box with Rounded Corners and Diagonals

You already know how to specify the width, height, and border thickness of a box. However, there are several features of this box that you have not seen before. This box has rounded corners, a diagonal, and different shading on either side of the diagonal. Here are the commands used to define this box:

DRAWBOX	2.87 in	1.5 in	14	SOLID			
command word	width	height	border thickness	border type			
	ROUNDED	ALL	DIAGONAL	RIGHT			
subcmd word	corner selection	subcmd word	diagonal selection				
	SHADE	LEFT	STANDARD	LIGHT	RIGHT	STANDARD	MEDIUM ;
shading option	shade area	shade pattern	shade type	shading option	shade pattern	shade type	end marker

Figure 155. **DRAWBOX** Subcommands

subcommand word

ROUNDED

corner selection

Specify which box corners you want rounded. Choose from:

ALL (Default)

TOPLEFT

TOPRIGHT

BOTTOMLEFT

BOTTOMRIGHT

You can choose **ALL** to round all corners of the box, or you can choose any combination of **TOPLEFT**, **TOPRIGHT**, **BOTTOMLEFT**, and **BOTTOMRIGHT** to select the corners individually. Specify more than one of these to round more than one corner.

subcommand word

DIAGONAL

More Features of DRAWBOX

diagonal selection

Choose from:

- LEFT** Defines the box to have a diagonal from the top-left corner to the bottom-right corner of the box.
- RIGHT** Defines the box to have a diagonal from the top-right corner to the bottom-left corner of the box.
- BOTH** Defines the box to have both, a left and right diagonal.

shading option **SHADE**

shade area You can choose which area of the box is to be shaded. Choose from:

- WHOLE** (Default)
- LEFT**
- RIGHT**
- TOP**
- BOTTOM**

Choose **WHOLE** to shade all of the box interior with the specified shading. This also applies to boxes with diagonals.

LEFT, **RIGHT**, **TOP**, and **BOTTOM** are specifiers for shading the portions of the box marked off by the diagonals. For example, in a box with a **LEFT** diagonal, **SHADE LEFT** shades the area to the left of the diagonal. However, in this example, **SHADE BOTTOM** shades the same area.

Note: **SHADE LEFT**, on a box with no diagonals, shades the whole box.

If you want a different shading in each area, simply repeat the shade area selection for each area you want shaded.

If a box has a dotted or dashed border and the box portions are shaded using **LEFT**, **RIGHT**, **TOP**, and **BOTTOM**, the gaps between the dashes or dots in the diagonal are not shaded. If shaded with **WHOLE**, the gaps in the diagonal are shaded. The gaps in the border are never shaded.

shade pattern Choose from:
STANDARD (Default)
SCREEN

Examples of **STANDARD** and **SCREEN** patterns in each available percentage of shade type are shown in Figure 211 on page 326 and Figure 212 on page 330 for examples of the two shade patterns.

shade type Customize shading by entering a number (*n*), for percentage of shading, or choose one of the ready-made shades. Choose from:

- XLIGHT**
- LIGHT**
- MEDIUM** (Default)
- DARK**
- XDARK**

n Percentage of shading, do not add the percent symbol (%).

end marker Always end a command with an end marker (;).

Shading with Dotted and Dashed Borders

If you use a border type of **DOTTED** or **DASHED** in conjunction with shading, the results depend on the way shading is specified.

More Features of DRAWBOX

Figure 156 shows the two possible results. In both examples, the shading only goes up to the inside edge of the border. In the example on the left, there is no shading between the dashes of the diagonal because the two parts of the box have been shaded separately (in the same way as Figure 154 on page 167). However, in the right-hand example, the spaces between the dashes have been shaded because the shading is **WHOLE**.

Note: Dots, and the spaces between them, have the same length as the border thickness. Dashes are 3 times the border thickness, and the spaces are equal to the border thickness. Therefore, the thicker the border the larger the dots and dashes.

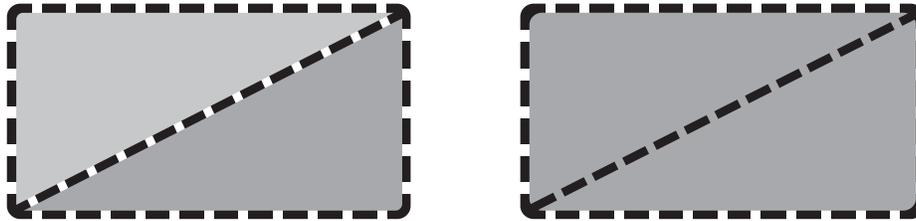


Figure 156. Shading Effects in Boxes

Keywords

Either the keyword or its abbreviation can be used in the overlay definition. Appendix I, “OGL/370 Keywords” on page 337 contains a complete list of all keywords and abbreviations that can be used in OGL/370.

More Features of DRAWBOX

Chapter 8. Additional Features for Circles and Paths

This chapter describes some additional features of the OGL/370 licensed program, which enable you to define circles and paths more easily. These features are presented in four sections:

- “More Features of **DRAWCIRCLE**” on page 173 explains more features of the **DRAWCIRCLE** command, such as specifying diagonals, shaded areas, dotted or dashed borders, and balanced and justified text.
- “Repeating Circles (**DRAWCIRCLE REPEAT**)” on page 175 explains how to repeat a circle by using the **REPEAT** subcommand in the **DRAWCIRCLE** command.
- “More Features of **DRAWPATH**” on page 179 explains more features of the **DRAWPATH** command, such as specifying path end shapes and rounded connections.
- “Repeating Paths (**DRAWPATH REPEAT**)” on page 186 explains how to repeat a path by using the **REPEAT** subcommand in the **DRAWPATH** command.

The only remaining command is **DRAWGRAPHIC**. **DRAWGRAPHIC** enables you to draw the same objects, and a few others, as you have previously learned (such as **DRAWCIRCLE**), but instead of generating MO:DCA statements, it generates Graphics Object Content Architecture (GOCA) statements. To use **DRAWGRAPHIC** for creating objects, see “**DRAWGRAPHIC** Command” on page 229.

When you finish this chapter, you will be able to write the definitions for any overlay OGL/370 can produce.

The commands in this chapter are for the sample overlay shown in Figure 157 on page 172, named “RESULT”. You have already seen how to define some of the figures on this overlay.

More Features of DRAWCIRCLE

This section describes other features of the **DRAWCIRCLE** command. These features include defining circles with balanced and justified text, diagonals, and shaded areas. It also explains how to adjust the appearance of dashed and dotted circles.

Figure 158 shows a circle with two diagonals and with two different shadings.

You already know how to specify the radius and border thickness of a circle.

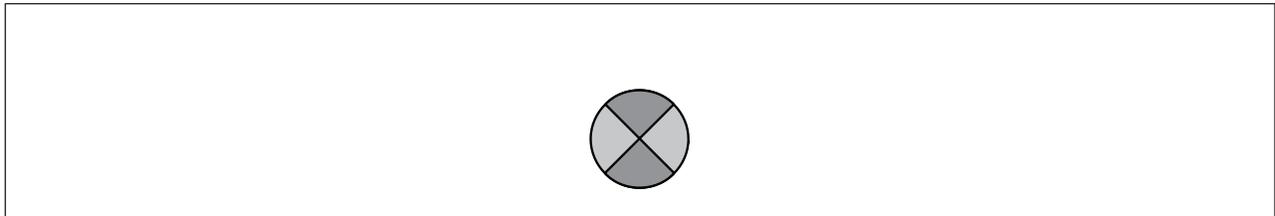


Figure 158. Sample Circle

This is how you define the circle in Figure 158:

DRAWCIRCLE	.5 in	BOLD	<u>SOLID</u>	<u>WHOLE</u>			
command word	radius	border thickness	border type	circle portion			
	DIAGONAL	BOTH	SHADE	LEFT RIGHT TOP BOTTOM	<u>STANDARD</u> <u>STANDARD</u> <u>STANDARD</u> <u>STANDARD</u>	LIGHT LIGHT <u>MEDIUM</u> <u>MEDIUM</u>	;
subcommand word	diagonal selection	option	shading	area	shade pattern	shade type	end marker

Figure 159. DRAWCIRCLE Subcommands

subcommand word

DIAGONAL

Note: Diagonals are only valid for whole circles.

diagonal selection

Choose from:

- LEFT** Creates a diagonal from the top-left corner to the bottom-right corner.
- RIGHT** Creates a diagonal from the top-right corner to the bottom-left corner.
- BOTH** Creates a right and left diagonal.

Shading: Describing shading for a circle is the same as describing shading for a box.

shading option **SHADE**

shade area

- Choose from:
- WHOLE (Default)
 - LEFT
 - RIGHT
 - TOP
 - BOTTOM

More Features of DRAWCIRCLE

shade pattern Choose from:
STANDARD (Default)
SCREEN

shade type Choose from:
XLIGHT
LIGHT
MEDIUM (Default)
DARK
XDARK

n Percentage of shading, do not enter the percent symbol (%).

Dotted and Dashed Borders

If you choose a border type of **DOTTED** or **DASHED**, you can also use the **AXIS** subcommand to specify the way in which the dots (or dashes) are arranged around the circle. However, this only has noticeable effect when the number of dots (or dashes) is less than about 15.

Figure 160 illustrates the effects produced by varying the value of the **AXIS** subcommand. **NONE** (default) causes the dots (or dashes) to be evenly distributed around the circle. **HORIZONTAL** causes them to be evenly distributed above and below the horizontal axis; **VERTICAL** causes them to be evenly distributed to the left and right of the vertical axis; and **BOTH** causes them to be distributed evenly within each quadrant.

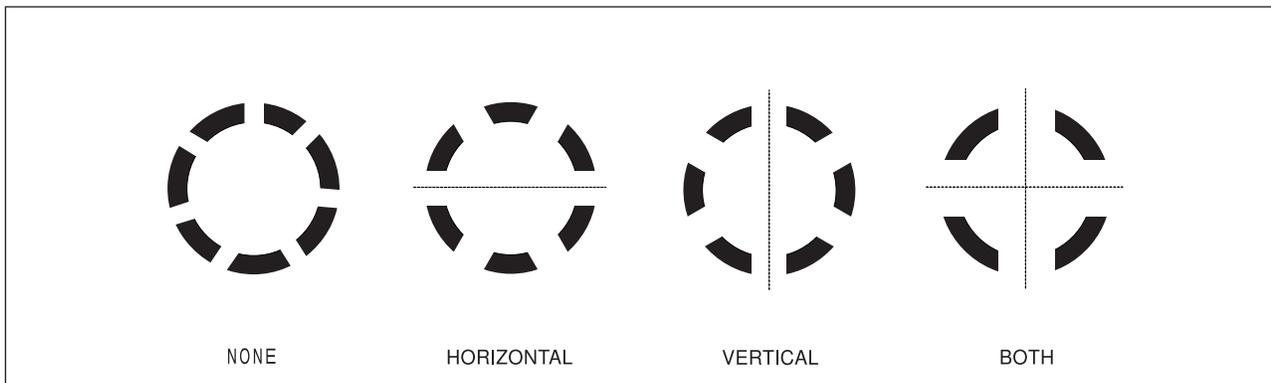


Figure 160. The effects of the **AXIS** subcommand

Note: Dots, and the spaces between them, have the same length as the border thickness. Dashes are three times the border thickness, and the spaces are equal to the border thickness. Therefore, the size of dots (or dashes) is proportional to the thickness of the border; conversely, the number of dots (or dashes) in a circle is inversely proportional to the thickness of the border.

Repeating Circles (DRAWCIRCLE REPEAT)

The **REPEAT** option for **DRAWCIRCLE** is similar to **DRAWBOX**. Repeated circles are the same size and have the same border thickness, yet they can have different shading and text.

Notice the two small circles (numbered 1 and 2) in Figure 161. Although we have already seen how to draw these circles separately in “Drawing Circles” on page 53, we could also have drawn them with one **DRAWCIRCLE** command using the **REPEAT** option.

As with rules and boxes, there are two ways to repeat circles:

Spaced Repetition: Enter an equal distance between circles.

Location Repetition: Enter the location of each circle.

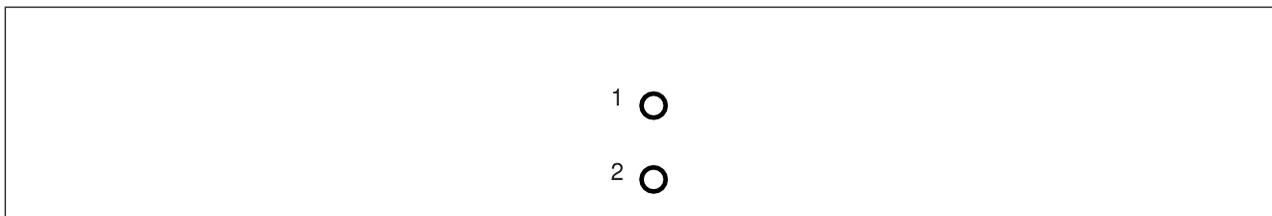


Figure 161. Repeated Circles

If you answer “yes” to both of the following questions, use spaced repetition; otherwise, you must use location repetition:

1. Are the centers of the circles lined up either vertically or horizontally?
2. Is the spacing between all the circles equal?

Looking at the circles in Figure 161, you can see that the answer to both of these questions is “yes” — the centers of the circles are lined up vertically, and since there are only two circles, the spacing is equal. Therefore, spaced repetition should be used for these circles.

The **REPEAT** subcommand follows the **DIAGONAL** subcommand (if you have chosen to specify it). Spaced and location repetitions are explained separately.

Spaced Repetition: The **REPEAT** subcommand has the following entries:

DRAWCIRCLE	0.06 in	10	<u>SOLID</u>	<u>WHOLE</u>	
command word	circle radius	border thickness	border type	circle portion	
REPEAT	DOWN	1	SPACED	0.35 in	;
subcmd word	direction	repetitions	spacing option	spacing value	end marker

Figure 162. **DRAWCIRCLE REPEAT** (Spaced Repetition)

subcommand word

REPEAT

direction

Refers to the direction the circles are repeated. Choose from:
ACROSS (Default)

Repeating Circles

DOWN

repetitions The number (*n*) of additional circles.

spacing word **SPACED**

spacing value horizontal distance (after **ACROSS**) or the vertical distance (after **DOWN**) between repeated circles. The unit of measurement can default to the **SETUNITS** value, or you can specify one of the following:

DIAMETER (Default)

n

IN Inches

MM Millimeters

PELS Pels

“Positioning the Circle (**POSITION**)” on page 55 shows a circle positioned by specifying where its center should be placed. The spacing between circles is measured from the center of one circle to the center of the next.

Notes:

1. The default spacing for circles is **DIAMETER**. This causes the circle centers to be spaced one diameter apart, which means that the diameters are just touching one another.
2. If you specify a spacing value of 0 for circles, the distance between the centers of the circles is 0. In other words, only one circle is visible, since the circles are placed in exactly the same position.

end marker Always end a command with an end marker (;).

Location Repetition: The **REPEAT** subcommand has the following entries:

DRAWCIRCLE	0.06 in	10	SOLID	WHOLE
command word	circle radius	border thickness	border type	circle portion
REPEAT	LOCATION	2.44 in	8 in	;
subcmd word	location option	horizontal coordinate	vertical coordinate	end marker

Figure 163. **DRAWCIRCLE REPEAT** (Location Repetition)

subcommand word

REPEAT

location option **LOCATION**

Identifies the type of repetition and indicates coordinates for the circle center that follows. You must enter this word and the coordinates for each repetition of the circle.

horizontal coordinate

A number (*n*) and unit of measurement that specify the horizontal coordinate of the center of the circle measured from the overlay origin.⁹ The unit of measurement can default to the **SETUNITS** unit, or you can specify one of the following:

n

9. If the command is part of a **GROUP** definition (see “Defining and Placing Groups” on page 162), the coordinates are measured from the group origin.

IN Inches
MM Millimeters
PELS Pels

vertical coordinate

A number (*n*) and unit of measurement that specify the vertical coordinate of the center of the circle measured from the overlay origin.⁹ The unit of measurement can default to the **SETUNITS** value, or you can specify one of the following:

n

IN Inches
MM Millimeters
PELS Pels

end marker Always end a command with an end marker (;).

Varying Shading and Text

Vary the shading or text that appears in individual circles created with **DRAWCIRCLE REPEAT** in the same way you vary shading or text in boxes created with **DRAWBOX REPEAT**.

To specify shading or text for individual circles created with **REPEAT**, you must know the number of the circle or circles to which the shading or text applies. Determining the circle number depends on how the circles are repeated.

Location Repetition: The circles are numbered beginning with 1 in the order in which you defined them. For example, in Figure 161 on page 175, the circles are numbered 1 and 2 according to the order in which their coordinates appear in the following command.

```
DRAWCIRCLE 0.06 IN 10 SOLID WHOLE -'DEFINES CIRCLE 1'
REPEAT LOCATION 2.44 IN 8 IN ; -'DEFINES CIRCLE 2'
```

Spaced Repetition: The circles are numbered beginning with 1 from left to right (for **ACROSS**) or from top to bottom (for **DOWN**). For example, in Figure 161 on page 175 the circles are numbered as OGL/370 sees them.

Varying Shading

After you have written the entire **REPEAT** subcommand, specify shading.

DRAWCIRCLE	0.06 in	10	SOLID	WHOLE	
command word	circle radius	border thickness	border type	circle portion	
REPEAT	DOWN	1	SPACED	0.35 in	
subcommand word	direction	repetitions	spacing option	spacing value	
SHADE	CIRCLE 1 CIRCLE 2	WHOLE WHOLE	STANDARD STANDARD	LIGHT LIGHT	;
shading option	circle	shade area	shade pattern	shade type	end marker

Figure 164. **DRAWCIRCLE REPEAT** (Shading)

The shading option has the following parts:

Repeating Circles

shading option **SHADE**

circle Specify the circle or circles to be shaded. Choose from:
ALL All the circles have the same shading (default.)
CIRCLE *n* All the circles do not have the same shading. For each circle to be shaded, enter **CIRCLE** followed by the circle number and the shade pattern and type.

shade area Specify the area of the whole or partial circle to be shaded.
WHOLE (Default)
LEFT
RIGHT
TOP
BOTTOM

Partial circles cannot have any diagonals in them. Therefore the only valid option for partial circles is **WHOLE**. This causes all of the partial circle to be shaded.

WHOLE refers to the entire interior of the circle, including circles with diagonals. If the circle has no diagonals, **WHOLE** is the only valid option.

shade pattern and shade type

Shade patterns and shade types are the same as those described in “Shading” on page 44.

For example, the two circles at the bottom of overlay **RESULT** have **STANDARD** pattern and **LIGHT** shading.

end marker Always end a command with an end marker (;).

Varying Text

If all the circles have the same text, you specify the text with one **WITHTEXT** subcommand. Otherwise, the text for each circle must be specified in a separate **WITHTEXT** subcommand. More than one **WITHTEXT** subcommands can be included within a single **DRAWCIRCLE** command.

The **WITHTEXT** subcommand appears after the **SHADE** subcommand, or if there is no **SHADE** subcommand, it appears after the **DIAGONAL** option.

The only change to the subcommand presented in “Repeating Boxes (**DRAWBOX REPEAT**)” on page 154 is that instead of the word **BOX** immediately after the subcommand word **WITHTEXT**, use the word **CIRCLE** as follows:

circle Specify the circle or circles in which text is to be included. Choose from:
ALL All the circles have the same text.
CIRCLE *n* All the circles do not have the same text. In each **WITHTEXT** subcommand, enter **CIRCLE** followed by the appropriate number.
CIRCLE 1 (Default)

More Features of DRAWPATH

This section describes other features of the **DRAWPATH** command, which include defining shading for closed paths and defining end shapes for open paths. Figure 165 illustrates paths on the overlay **RESULT**.

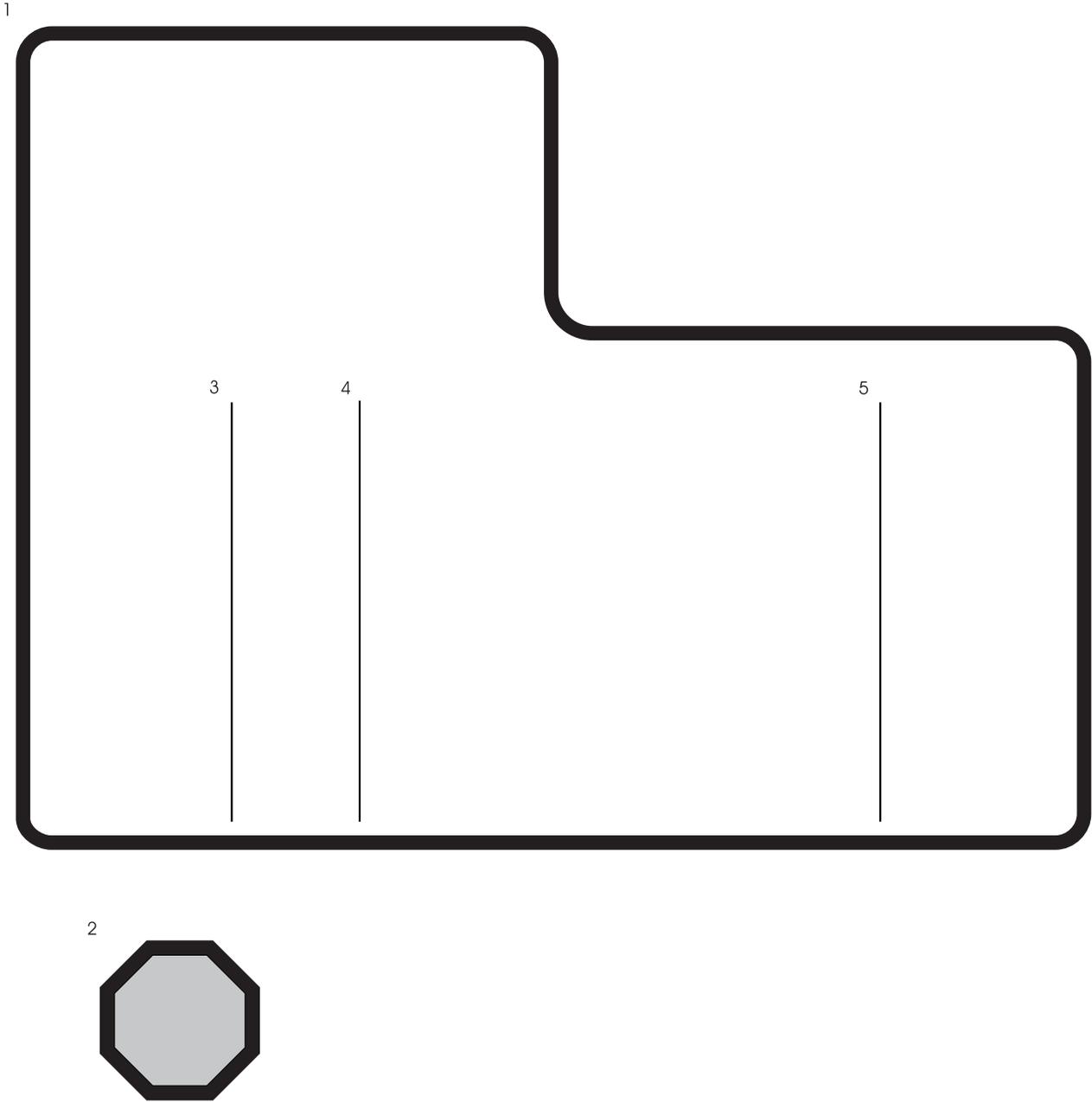


Figure 165. Extract from Overlay **RESULT** Used to Illustrate Paths

Look at Path 2 in Figure 165. This path is different from those you have already seen. It is made up of horizontal, vertical, and diagonal sides and is shaded. You already know how to specify the path shape, border thickness, and border style. Figure 166 on page 180 illustrates the command used to define Path 2.

More Features of DRAWPATH

DRAWPATH	10	<u>SOLID</u>	CONNECTION	<u>MITER</u>
command word	path thickness	path type	subcommand word	connection type
TO	<u>ABSOLUTE</u>	1.49 in	<u>ABSOLUTE</u>	7.56 in
TO	<u>ABSOLUTE</u>	1.7 in	<u>ABSOLUTE</u>	7.77 in
TO	<u>ABSOLUTE</u>	1.7 in	<u>ABSOLUTE</u>	8.07 in
TO	<u>ABSOLUTE</u>	1.49 in	<u>ABSOLUTE</u>	8.28 in
TO	<u>ABSOLUTE</u>	1.19 in	<u>ABSOLUTE</u>	8.28 in
TO	<u>ABSOLUTE</u>	0.98 in	<u>ABSOLUTE</u>	8.07 in
TO	<u>ABSOLUTE</u>	0.98 in	<u>ABSOLUTE</u>	7.77 in
TO	<u>ABSOLUTE</u>	1.19 in	<u>ABSOLUTE</u>	7.56 in
subcmd word	origin option	first coordinate	origin option	second coordinate
SHADE	<u>STANDARD</u>	LIGHT	;	
shading option	shade pattern	shade type	end marker	

Figure 166. DRAWPATH Command

To shade the path, specify the following:

shading option **SHADE**

Note: Only closed paths can be shaded. These are paths which start and end on the same point on your overlay. Make a closed path by placing the end at the same point on the overlay as the start — as in Figure 166. You can use the **CLOSE** subcommand—as in Figure 53 on page 60.

shade pattern Choose from:

XLIGHT
LIGHT
MEDIUM (Default)
DARK
XDARK

n Percentage of shading, do not enter the percent symbol (%).

Figure 211 on page 326 and Figure 212 on page 330 show examples of the shade types.

Use the **DRAWPATH** command to define open paths. Open paths do not start and end on the same point on the overlay. When you define an open path, you specify the shape of the end points of the path. Figure 167 on page 181. shows an open path with rounded ends.

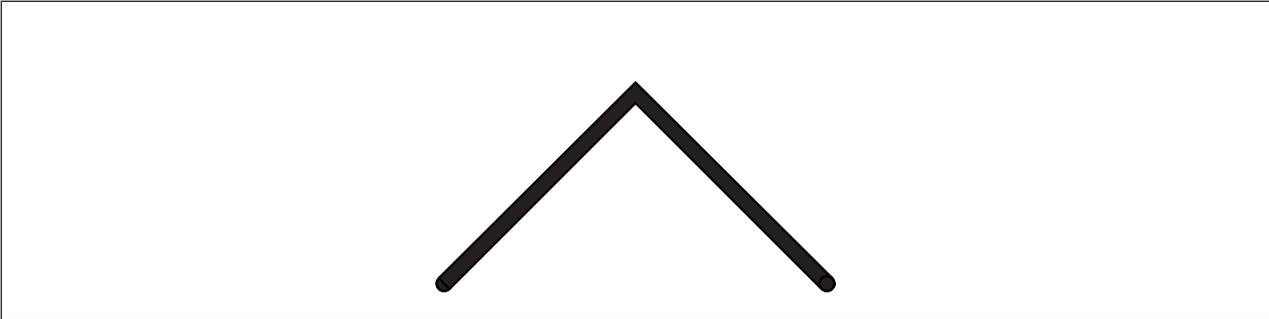


Figure 167. Open Path with Rounded Ends

The command used to draw the path illustrated in Figure 167 is:

DRAWPATH	40	SOLID	PATHEND	ROUNDED	ROUNDED
command word	path thickness	path type	subcommand word	path start shape	path end shape
TO	UP	1 in	RIGHT	1 in	
TO	DOWN	1 in	RIGHT	1 in	;
subcmd word	origin option	first coordinate	origin option	second coordinate	end marker

Figure 168. PATHEND Subcommand

To define the end shapes, specify the following:

subcommand word

PATHEND

path start shape

Select the shape of the start of the path from the following:

- TRIMMED**
- SQUARE**
- ROUNDED**

path end shape

Select the shape of the end of the path from the following:

- TRIMMED**
- SQUARE**
- ROUNDED**

Note: If you specify only one parameter here, it applies to both the path start and path end. Path end shapes can only be specified for open paths. Figure 169 on page 182 illustrates three types of end shapes.

More Features of DRAWPATH

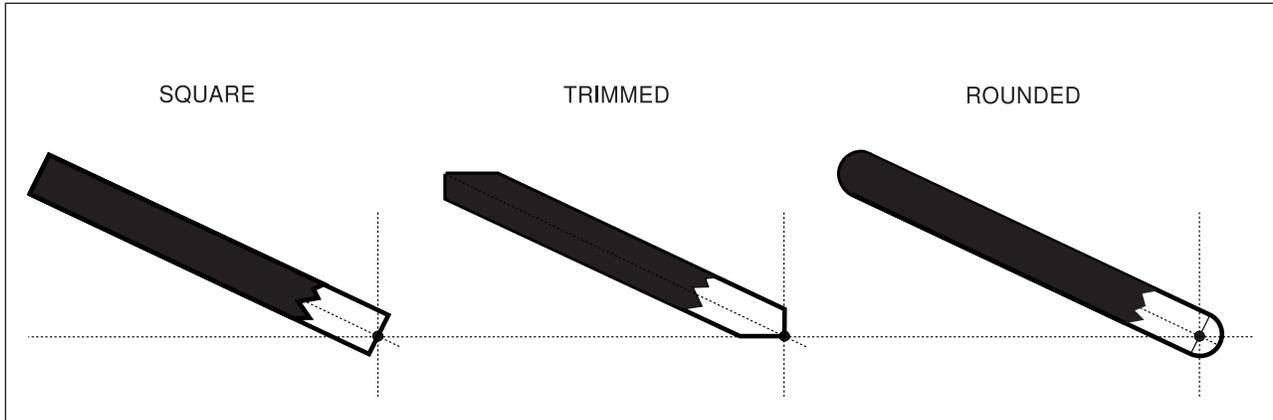


Figure 169. Path End Shapes

Trimmed Path Ends

The primary purpose of using a trimmed end on a path is so the end of a diagonal path can fit into the corner of a box even if the borderweight of the box is much less than the borderweight of the diagonal. See Figure 170.

If a path segment is too close to the horizontal or vertical position you may not get the results you expect. As a segment gets closer to horizontal or vertical, more is trimmed off one side and less off the other. Eventually, all of one side is trimmed off.

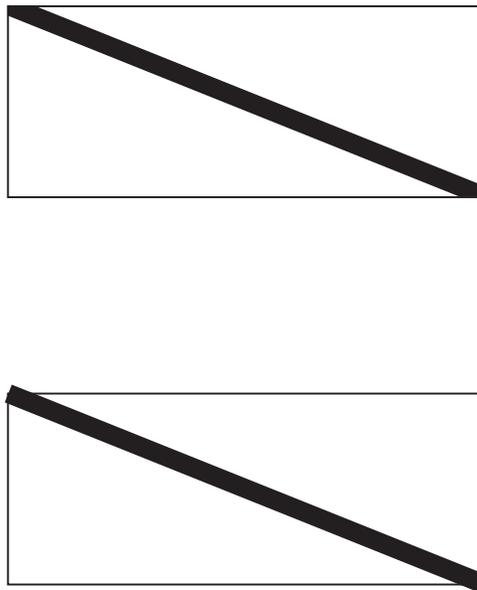


Figure 170. Examples of Path End Shapes in a Box. The figure on the top shows a **TRIMMED** path end and the figure on the bottom shows a **SQUARE** path end.



Figure 171. Examples of **TOPLEFT** and **CENTER** Positioning. The figure on the left shows **CENTER** positioning and the figure on the right shows **TOPLEFT** positioning.

If your box is dotted or dashed, you should be aware of the **TOPLEFT** versus **CENTER** positioning option in the **SETUNITS** command. If your diagonal path segment is positioned at the same points as the corners of the box, the diagonal overlaps all or part of the box border, depending on that option. See Figure 171.

OGI/370 does not extend any trimming into a connection between path segments. For example, if a trimmed segment is connected to another segment by a large rounded corner, and the rounded corner is too close to the trimmed end, you may see a bump where the trimmed segment meets the untrimmed rounded connection.

Length of Rounded Connections

Chapter 3, “Drawing Circles and Paths” on page 53, describes how to define a path with rounded connections. Each rounded connection is an arc of a circle. The corner length of a rounded connection is the distance between the corner point and the point on the straight line where the corner arc begins. Figure 130 on page 142 illustrates how you determine the exact corner length of a rounded connection.

The default corner length used when drawing paths with rounded connections is defined in the last **SETUNITS** command (or is **MEDIUM** if no **SETUNITS** has been used). You can also define the corner length for all connections of a path using the **CONNECTION** subcommand of **DRAWPATH**. This overrides the **SETUNITS** default.

Define the corner length of a single connection using the **CONNECTION** option of the **TO** subcommand.

When you define the corner length for a single connection, this overrides the **SETUNITS** default, and any value set for the path. Figure 172 on page 184 shows a path with two different corner lengths at different connections.

More Features of DRAWPATH



Figure 172. Two Corner Lengths

To do this, provide a corner length for each connection.

command word	path weight	path type							
DRAWPATH	60	SOLID							
TO	UP	1.25 in	RIGHT	0 in	CONNECTION	ROUNDED	0.25 in		
TO	UP	0 in	RIGHT	2.75 in	CONNECTION	ROUNDED	0.5 in		
TO	DOWN	1.25 in	RIGHT	0 in	;				
subcmd word	origin option	first coordinate	origin option	second coordinate	subcommand word	connect type	connect length	end marker	

Figure 173. **DRAWPATH** Command with Corner Lengths

To define corner length, specify the following:

subcommand word

CONNECTION

connection type

ROUNDED

connection length

The corner length keywords are:

SMALL The corner length used is 10 pels.

MEDIUM The corner length used is 20 pels.

LARGE The corner length used is 30 pels.

MAX The corner length used is equal to the shorter of the two path segments being connected. For example, if the two path segments are at right angles, the radius of the arc forming the rounded connection is equal to length of the shorter path segment.

HALF The corner length used is equal to half the shorter of the two path segments being connected. For example, if the two path segments are at right angles, the radius of the arc forming the rounded connection is equal to half the length of the shorter path segment.

n

IN Inches
MM Millimeters
PELS Pels
BW Border weight

More Features of DRAWPATH

end marker Always end a command with an end marker (;).

Using one of the corner length keywords should satisfy most requirements. Experiment with different corner lengths and decide which one best suits your overlay.

As mentioned before, you specify the corner length for a whole path. To do this, use the **CONNECTION** subcommand before the first **TO** subcommand. Override this value for particular connections.

The example below would also generate Figure 172 on page 184:

```
DRAWPATH 60 SOLID CONNECTION ROUNDED 0.25 IN
TO UP 1.25 IN RIGHT 0 IN
TO UP 0 IN RIGHT 2.75 IN CONNECTION ROUNDED 0.5 IN
TO DOWN 1.25 IN RIGHT 0 IN ;
```

Figure 174. **DRAWPATH** Command with Corner Lengths

Repeating Paths

Repeating Paths (DRAWPATH REPEAT)

As with rules, boxes, and circles, there are two ways to repeat paths:

Spaced Repetition: Enter an equal distance between path origins.

Location Repetition: Enter the location of each path origin.

If you answer “yes” to both of the following questions, use spaced repetition; otherwise, you must use location repetition:

1. Are the path origins lined up either vertically or horizontally?
2. Is the spacing between all the paths equal?

When we look at the paths in Figure 175, we can see that the answer to the first question is “no” — the origins of the paths are not lined up vertically or horizontally.

Therefore, location repetition is appropriate for these paths.

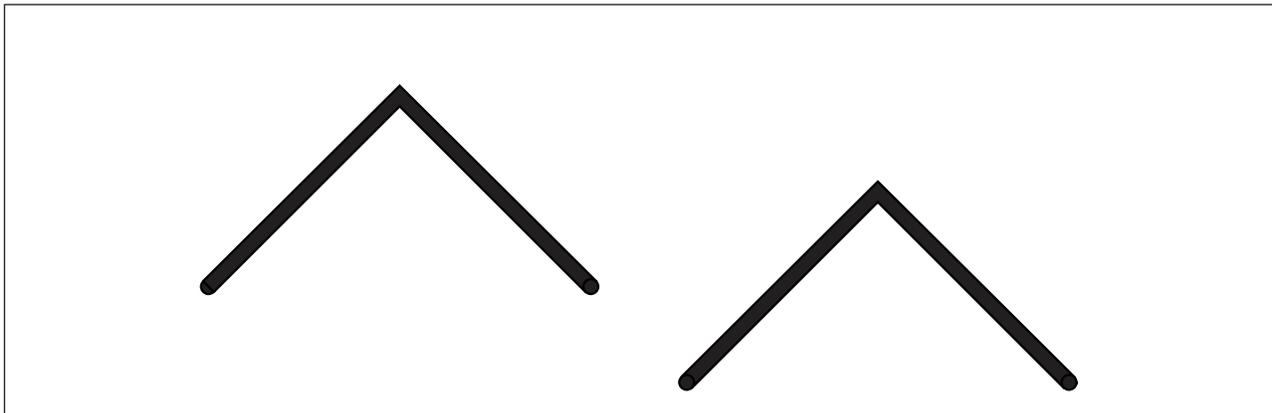


Figure 175. Repeated Paths

The **REPEAT** subcommand follows the **CONNECTION** subcommand, if you have specified it. Spaced and location repetitions are explained separately.

Spaced Repetition: The **REPEAT** subcommand is similar to that for circles. See “Repeating Circles (**DRAWCIRCLE REPEAT**)” on page 175.

The options for spacing value with **DRAWPATH** are:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: There is no default value for spacing with **DRAWPATH REPEAT**. However, you can use the default unit of measurement specified in the previous **SETUNITS** command.

Location Repetition: Location repetition for paths is similar to location repetition for circles, explained in “Repeating Circles (**DRAWCIRCLE REPEAT**)” on page 175.

The options for spacing value with **DRAWPATH** are:

n

IN	Inches
MM	Millimeters

PELS Pels

Varying Shading

Vary the shading that appears in individual paths created with **DRAWPATH REPEAT**, in the same way that you vary shading in circles created with **DRAWCIRCLE REPEAT**. Only closed paths can be shaded.

To specify shading for individual paths created with **REPEAT**, you need to know the number of the path or paths to which the shading applies. Determining the path number depends on how the paths are repeated.

Location Repetition: The paths are numbered beginning with 1 in the order in which they were defined.

Spaced Repetition: The paths are numbered beginning with 1 from left to right (for **ACROSS**) or from top to bottom (for **DOWN**).

After you have written the entire **REPEAT** subcommand, specify shading. The shading option has the following parts:

shading option **SHADE**

path Specify the path or paths to be shaded. Choose from:

ALL All the paths have the same shading (default).

PATH *n* All the paths do not have the same shading. For each path to be shaded, enter **PATH** followed by the path number and the shade pattern and type.

shade pattern and shade type

end marker Always end a command with an end marker (;).

Repeating Paths

Part 4. Reference

Chapter 9. Overlay Generation Language Commands

Syntax Rules for Commands

General Syntax Rules

- Do not use column 1 when writing commands.
- Leave a space before and after each part of a command. The only exception to this rule is the semicolon (;) that ends each command. You do not have to leave a space before or after a semicolon used to end a command.
- Except for comments, which you may use at any point in your overlay definition, enter the parts of commands in the same sequence as they appear in this manual.
- End each command with a semicolon (;). OGL/370 ignores anything written on a line after a semicolon.
- OGL/370 recognizes the commands, subcommands, keywords, and names you enter even if you mix uppercase and lowercase letters.

Note: The exception to this rule is text, where all entered characters appear just as you type them.

Numbers in Commands

- Leave a space before and after numbers.
- Use positive numbers in decimal form and use only one decimal point.
- Use only numbers between 0 and 9999.99.
- Make sure the numbers you use go only two places to the right of the decimal point.
- Do not mix letters, punctuation, or symbols with numbers.

Comments in Your Command Statements

- In most cases, a single comment can extend over more than one line. However, if a comment appears on a line after the end-marker semicolon (;), you should not continue the comment on the next line. You can, of course, begin a new comment on the next line.
- You can make single-word comments or block comments anywhere in a command.
- Leave a space before and after each single-word comment or block comment.
- Use a hyphen (-) as the first character in each single-word comment.
- Use a hyphen and an apostrophe (-') as the first two characters of a block comment.
- End a block comment with an apostrophe (').

Note: When your block comment contains an apostrophe or a semicolon, type two apostrophes (") to show the apostrophe and two semicolons (;;) to show the semicolon.

Text in Overlays

- An apostrophe (') must precede the first character and follow the last character of each text segment. (as in the **SETTEXT** command).
- When your text contains an apostrophe or a semicolon, type in two apostrophes (") or two semicolons (;;), respectively.

Abbreviations for Units of Measurement

The shortest abbreviations for units of measurement used in overlays are:

- Inches (**IN**)
- Millimeters (**MM**)
- Pels (**PELS**)
- Lines per inch (**LPI**)
- Characters per inch (**CPI**)

Syntax Rules for Commands

Border weight (**BW**)

Command Syntax Diagrams

The figures in this chapter, which contain syntax diagrams are used to illustrate command words.

Reading Syntax Diagrams

The syntax for OGL/370 commands is shown using graphic notation. To read the diagrams, move from left to right and top to bottom, following the main path line.

Style Rules

Syntax diagrams use the following style rules to show how to enter commands and parameters:

- A word in all italic, lowercase letters shows a parameter that you replace. For example:

member ID

shows that you replace *member ID* with an identifier such as "PALM2".

- A parameter above the line shows the default parameter. For example, **NOTRACE** is the default parameter in the syntax diagram for the **CBDUMP** command:

CBDUMP Command



Symbols

Syntax diagrams use symbols to help you follow the flow of information they communicate.

- Statements begin with:

▶▶

and end with:

→▶

- Statements longer than one line continue to a second line with:

→

where they resume with:

▶

Required Parameters

A parameter that you must include is displayed on the main path line. For example, the syntax diagram for the **DEFINE** command:

DEFINE Command



shows that you must follow **DEFINE** with its required parameter.

If there are two or more required parameters from which to choose, the parameters are shown with the first choice on the main path line and the other choices on branch lines under it. For example, the syntax diagram for the **PLACE** Command:

PLACE Command



shows that you must type the command in any of the following ways:

```
PLACE SEGID
PLACE GROUP
```

Optional Parameters

Parameters that you can include with a command are displayed on branch lines below the main path line. For example, the syntax diagram for the **DRAWPATH** command:

DRAWPATH Command



shows that you can type the command in one of these ways:

```
DRAWPATH .1 IN
DRAWPATH 1 MM
DRAWPATH 2 PELS
```

Branch lines can include branch lines of their own. An example of this is the partial syntax diagram for the **DRAWBOX** command with the option **DIAGONAL** parameter:

DRAWBOX Command



Repeating Parameters

An arrow on a line above a parameter means that you can repeat the parameter, or enter more than one of the listed parameters. An example of this is the syntax diagram for **DEFINE** command:

DEFINE Command



The arrow above *coded line* means you can include one or more coded line parameters with the **DEFINE** command.

CBDUMP Command

CBDUMP Command

This command, which is used for diagnosing problems in the OGL/370 program, is not used to design or produce overlays. It is explained in *OGI/370: Diagnosis Guide and Reference*.

CBDUMP Command



command word

CBDUMP

trace option Allows you to specify the start of module tracing. Choose from:

NOTRACE (Default)

TRACE Traces all operations, except those which involve the following modules:
DZILSFMN
DZILSFCL
DZILRANP
DZILADDP

If tracing is required, this option is recommended because of the frequency with which the above modules are used.

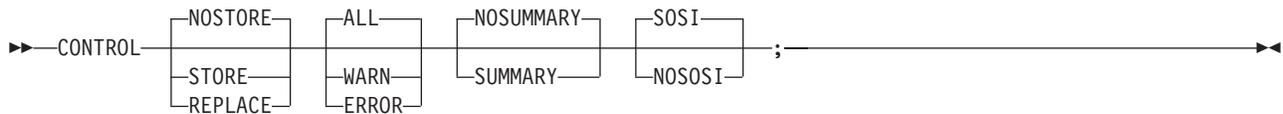
TRACEALL Traces all operations.

end marker Always end a command with an end marker (;).

CONTROL Command

Use this command to change the storage and message defaults for the overlay you are creating. You do not need to issue this command unless you want to change existing default values for messages and overlay storage. Each overlay can contain only one **CONTROL** command.

CONTROL Command



command word

CONTROL

storage

Specify the storage option for the overlay you are creating. The program produces a sample copy of the overlay if you select any of the options or if you allow the command to default to **NOSTORE**. Choose from:

NOSTORE After the sample copy is produced, the overlay is not stored in a library (default).

STORE After the sample copy is produced, the overlay is stored in a library under the name you specify in the **OVERLAY** command. If an overlay of the same name already exists in the library and you specify **STORE**, the program issues an error message and the overlay is not stored.

REPLACE After the sample copy is produced, the overlay is stored under the name specified in the **OVERLAY** command regardless of whether an overlay with that name already exists.

Note: When you specify **STORE** or **REPLACE**, the program adds the prefix “O1” to the overlay name (specified in the **OVERLAY** command) and stores the overlay in a library under the full name.

messages

Specify the type of messages you want to receive:

ALL To see every message the program issues while processing your overlay (default).

WARN To see only the warning and error messages that result from your overlay’s processing.

ERROR To see only the error messages issued as a result of OGL/370 processing.

Notes:

1. Messages appear on the source listing for your overlay.
2. See “Messages” on page 6 for a complete description of message

summary

If you want OGL/370 to compile a summary of statistics about your overlay.

NOSUMMARY

A summary does not appear as part of the source listing (default).

SUMMARY

A statistical summary appears on the source listing.

Note: For a complete explanation of this summary, see Appendix J, “Storage Summary” on page 341.

sosi option

Specifies the way SOSI (shift out/shift in) delimiters are to be handled. Choose from:

CONTROL Command

SOSI Double-byte text must be surrounded by SOSI delimiters.

NOSOSI

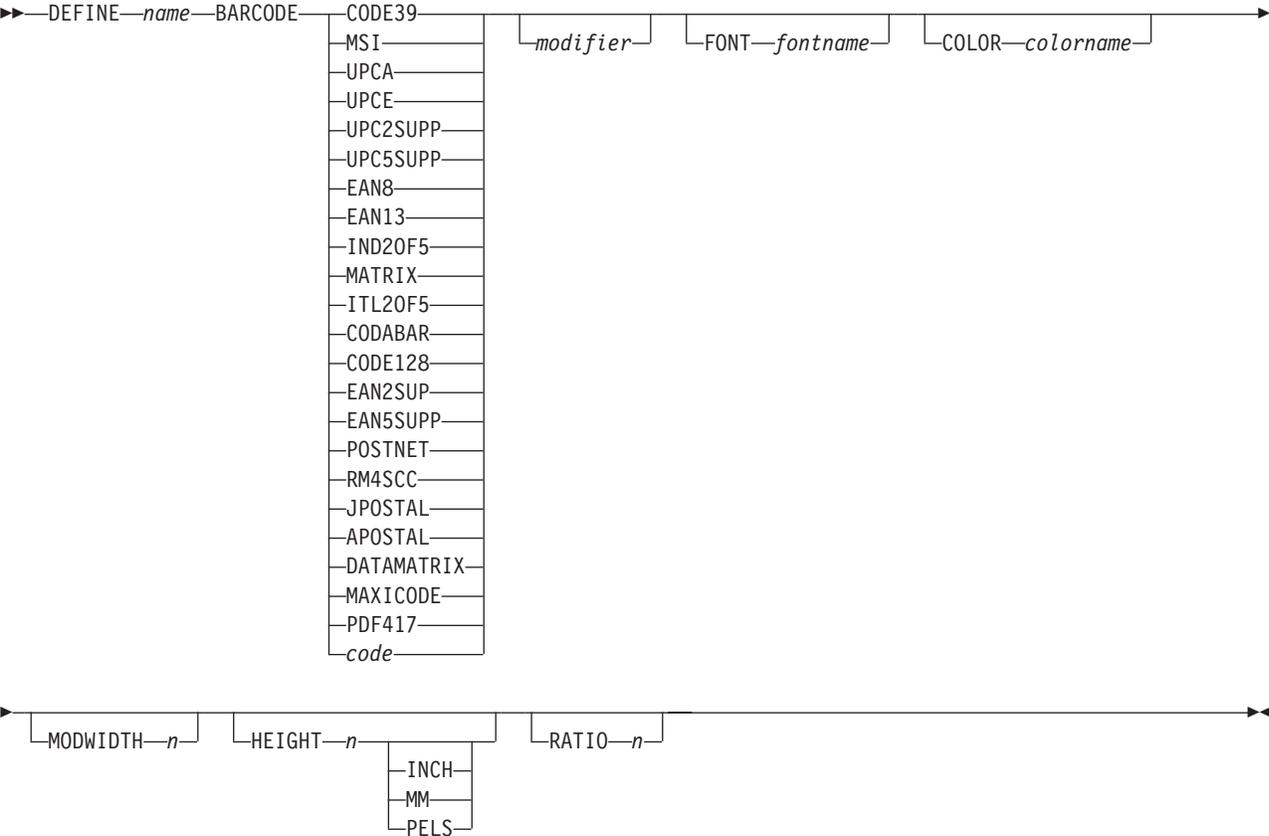
Double-byte text **must not** be surrounded by SOSI delimiters.

end marker Always end a command with an end marker (;).

DEFINE BARCODE Command

With the **DEFINE BARCODE** command, you can define a barcode symbology for use in **PLACE-BARCODE** commands to put BCOCA barcodes on your overlay.

DEFINE — BARCODE Command



command word

DEFINE

name The name of the barcode definition.

BARCODE Identifies this **DEFINE** statement as a barcode definition.

type Defines the BCOCA symbology type. Either use one of the following keywords:

- CODE39**
- MSI**
- UPCA**
- UPCE**
- UPC2SUPP**
- EAN8**
- EAN13**
- IND2OF5**
- MATRIX**
- ITL2OF5**
- CODABAR**
- CODE128**
- EAN2SUP**
- EAN5SUPP**

DEFINE BARCODE Command

| **POSTNET**
| **RM4SCC**
| **JPOSTAL**
| **APOSTAL**
| **DATAMATRIX**
| **MAXICODE**
| **PDF417**

| or enter the BCOCA code value. See *Data Stream and Object Architecture: Bar Code Object Content Architecture (BCOCA) Reference*, S544-3766-04 for the code values. If you choose a symbology that does not match a keyword, OGL issues a warning message and continues with reduced parameter checking.

| **Note:** Selecting unsupported 2D symbologies will result in incorrect output.

| *modifier* Defines the modifier associated with the *type*.

| If *modifier* is not specified, the default is the minimum value for the *type*. See *Data Stream and Object Architecture: Bar Code Object Content Architecture (BCOCA) Reference*, S544-3766-04 for the range of values for each *type*.

| **FONT** Specifies the name of the font for any HRI characters. This **FONT** parameter must also be defined in the **FONT** command.

| If **FONT** is not specified, the default value is the default for the symbology.

| **COLOR** Specifies the name of the color used for the symbols and any HRI characters. This **COLOR** parameter must also be defined in the **DEFINE COLOR** command.

| If **COLOR** is not specified, the default value is the default for the printer.

| **MODWIDTH** Specifies the width of the narrowest bar used by the symbology. **MODWIDTH** must be an integer ranging from 1 and 254. Values in this range that are *not* supported by the defined symbology may result in an error.

| If **MODWIDTH** is not specified, the default value is the default for the symbology.

| **HEIGHT** Specifies the height of the symbols. Values that are *not* supported by the defined symbology may result in an error. With a number (*n*) and a unit of measurement, specify the height of your barcode. Choose from:

| *n*
| **IN** Inches
| **MM** Millimeters
| **PELS** Pels

| If *barcode height* is not specified, the default value is the default for the symbology. If *units* is not specified, the default value is the from the **SETUNITS** command.

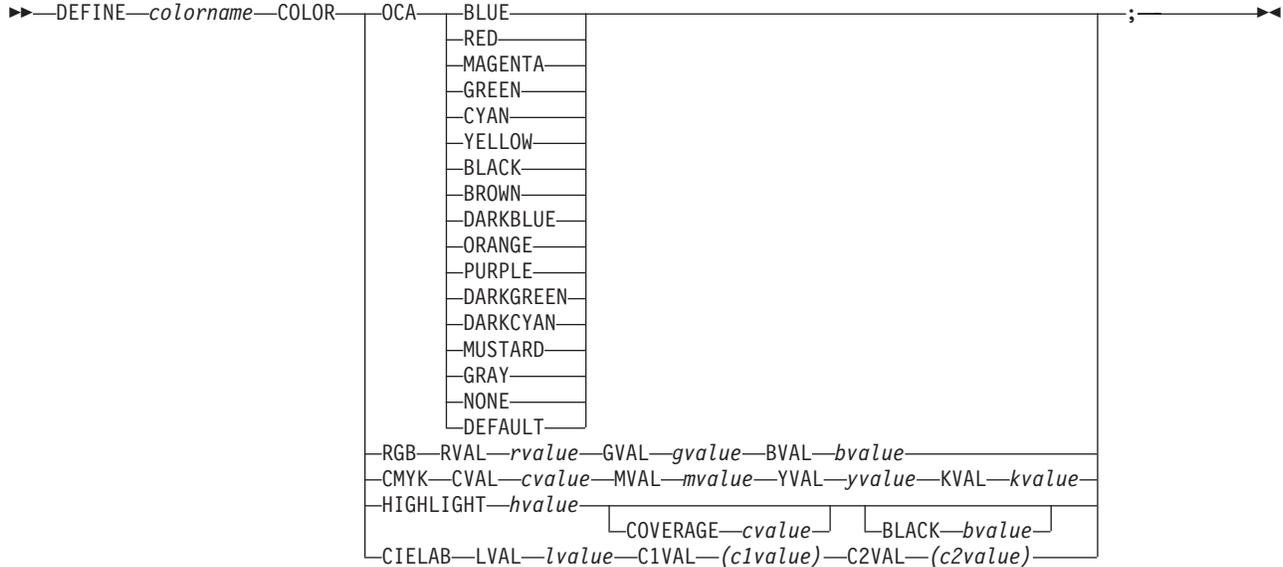
| **RATIO** Specifies the ratio of the wide bar to the narrow bar. The value for **RATIO** must be an integer ranging from 1.01 to 9.99. Values in this range that are *not* supported by the defined symbology may result in an error.

| If **RATIO** is not specified, the default value is the default for the symbology.

DEFINE COLOR Command

With the **DEFINE COLOR** command, you can identify a color using one of five color models.

DEFINE — COLOR Command



command word

DEFINE

colorname The name of the color definition. The name is then used on subsequent **DRAWRULE**, **DRAWMASK**, **SETTEXT**, **WITHTEXT**, **DRAWBOX**, **FONT**, or **PLACE PATTERN** commands. *colorname* can only include the characters A—Z, 0—9, @, #, -, and \$. The first character cannot be a hyphen. The name cannot be longer than 8 characters nor can it be a reserved word such as **RED**, **BLUE**, and so forth.

COLOR Identifies this **DEFINE** statement as a color definition. Should be followed by one of the valid color model keywords such as **OCA**, **RGB**, and so forth.

OCA Defines the IBM OCA color model. Should be followed by one of the valid color keywords such as **RED**, **BLUE**, and so forth. **NONE** specifies that no color is associated with the object. This is the same as *color of medium*. If text with `color=none` is placed on top of a box with blue background color, the pels in the text erase the blue pels. The **DEFAULT** keyword means *Presentation Process Default* which is the printer's primary toner or ink color. There is no default color for the OCA parameter.

RGB Defines the **RGB** (Red/Green/Blue) color model. Each value is specified as a percentage from 0 to 100. *Rvalue* specifies the red value, *Gvalue* specifies the green value, and *Bvalue* specifies the blue value.

An **RGB** specification of 0/0/0 is black, 100/100/100 is white, and any other value is color somewhere in between, depending on the output device.

RVAL, **GVAL**, and **BVAL** must be specified in this order. Fractional values are ignored.

CMYK Defines the *cyan/magenta/yellow/black* color model. Each value is specified as a percentage from 0 to 100. *Cvalue* specifies the cyan value, *Mvalue* specifies the magenta value, *Yvalue* specifies the yellow value, and *Mvalue* specifies the black value.

CVAL, **MVAL**, **YVAL**, and **KVAL** must be specified in this order. Fractional values are ignored.

DEFINE COLOR Command

HIGHLIGHT Defines the highlight color model. Highlight colors are specific to the IBM InfoPrint Hi-Lite Color Post Processor. *Hvalue* is within the range 0 to 65535. An *hvalue* of zero equates to the presentation device default color. A non-zero value should match the value of the color defined at the printer. There is no default subvalue. Fractional values are ignored.

BLACK Indicates the percentage of black to add to the highlight color. *Bvalue* is a percentage within the range 0 to 100. The amount of black shading applied depends on the **COVERAGE** percentage, which is applied first.

If **BLACK** is not specified, the default value of 0 is used.

CIELAB Defines the **CIELAB** model. *Lvalue* specifies the luminance value and is within the range 0.00 to 100.00. *C1value* and *C2value* specify the chrominance differences and are specified as signed integers from -127 to 127. The parentheses are required.

LVAL, **C1VAL**, and **C2VAL** must be specified in this order. There are no defaults for the subvalues.

Note:

- You must know the capabilities of your printer before using this function. Some printers restrict color and percent shading to certain areas on the page.
- When using **DEFINE COLOR**, do not specify **ENDDEF**. The **ENDDEF** is ignored.
- Do not put **DEFINE COLOR** within a **DEFINE GROUP**, but the valid commands within a group definition (for example, **DRAWBOX**) can refer to color definitions.
- Define the color before calling out the color definition on **DRAWMASK**, **DRAWRULE**, and so forth.
- Do not use keywords like **BLUE** or **MAGENTA** as the nickname for the **DEFINE COLOR** definition.
- The **COVERAGE** plus **BLACK** values cannot exceed 100 percent. If **COVERAGE** plus **BLACK** adds up to less than 100 percent, the remaining percentage is achieved with *color of medium*.
- If text defined with a non-zero **COVERAGE** value is placed on top of an area with the same non-zero coverage value, the text appears to be invisible.
- On **DRAWBOX** and **PLACE PATTERN** the **SHADE** and **COLOR** parameters are mutually exclusive. If both are specified, the **SHADE** specification is ignored.

DEFINE GROUP Command

With the **DEFINE GROUP** command, you can develop a group of statements that describes a section of the overlay that you can place anywhere you wish as many times as you wish.

DEFINE — GROUP Command



command word

DEFINE

section name Section names must meet these requirements:

- The name can include only the following characters: A—Z, 0—9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than eight characters.

section type Use the **GROUP** (default) keyword to combine **DRAWBOX**, **DRAWCIRCLE**, **DRAWGRAPHIC**, **DRAWMASK**, **DRAWPATH**, **DRAWRULE**, **POSITION**, **SETTEXT**, and **SETUNITS** commands to create an overlay section for placement in your overlay.

Note: When you develop a **SETUNITS** command for use in a **GROUP**, the values you establish in that **SETUNITS** command are defaulted to only by the commands within the **GROUP**.

end marker Enter an end marker (;) immediately after entering the **GROUP** keyword.

group commands

Specify the command name, subcommands, and options for each command that the program must use to create the overlay section you want placed in this overlay.

Note: Remember that absolute positions are relative to the group origin.

end markers End each command statement with an end marker (;).

command word

This word must follow the last statement of a group of command statements:

ENDEF ;

end marker Follow the **ENDEF** command word with an end marker (;).

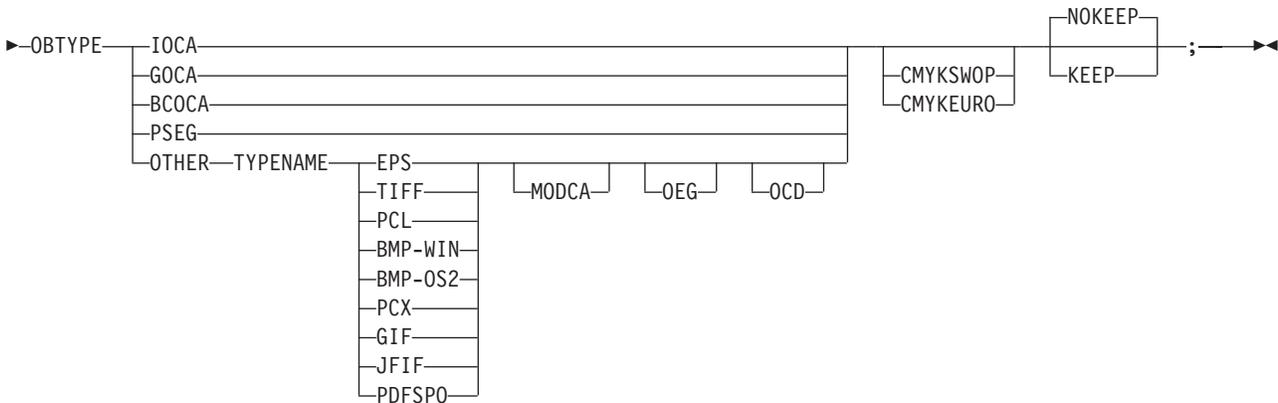
DEFINE OBJECT Command

DEFINE OBJECT Command

Use **DEFINE OBJECT** prior to the **PLACE OBJECT** command to establish a definition for the included object.

DEFINE — OBJECT Command

▶▶—DEFINE—*name*—OBJECT—OBNAME—*objmember*—▶▶



name The name of the object definition. The name is used on subsequent **PLACE OBJECT** commands. *name* can only include the characters A—Z, 0—9, @, #, -, and \$. The first character cannot be a hyphen. The name cannot be longer than 8 characters. This parameter is required.

OBJECT Identifies this **DEFINE** statement as an object definition.

OBNAME The file or member name of the object that should be included by PSF or the viewer. The name can be 1-8 characters. This parameter is required.

OBTYPE Pick one of the valid values. This parameter is required. **PSEG** means a page segment containing **IOCA** or **GOCA** image. The **OBTYPE** cannot be a page segment with IM1 image.

When **OBTYPE** is **OTHER**, the user must specify **TYPENAME** and provide the appropriate keyword to indicate whether the object is **EPS**, **TIFF**, and so forth.

When **OBTYPE** is **PSEG** and the page segment object name starts with 'S1', include the 'S1' in the **OBNAME**.

| **color profiles** Identifies the CMYK color profile to be used. The choices are:
| **CMYKSWOP** US CMYK color profile.
| **CMYKEURO** European CMYK color profile.

| **object retention**
| Indicates whether to keep the object in the printer for the whole time the overlay is being
| printed. The choices are:
| **NOKEEP** Used to reload the object each time it is included in the overlay (default).
| **KEEP** The object is included overlay making a hard object at the beginning of
| the page and then available throughout without reloading.

Note:

- Do not put **DEFINE OBJECT** or **PLACE OBJECT** within a **DEFINE GROUP**.

DEFINE OBJECT Command

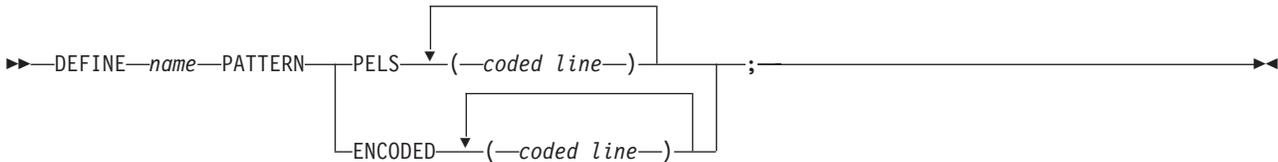
- If the object definition on **OBJECT** is not found, an error message is issued and the object is not included.
- The associated subparameters (**MODCA**, **OCD**, and **OEG**) are optional. When object type is **OTHER**, if you know the object has **MODCA** container records, specify the **MODCA**, **OCD**, or **OEG** keywords as appropriate. If you do not know whether the object contains these records, do not specify these keywords. If you specify only **OEG** or **OCD**, it is assumed the object is contained within **MODCA** wrappers. If you specify **MODCA** and **OEG** it is assumed the object also has an **OCD** record.
- Different presentation systems have differing levels of support for included object types.
- **FILL** on **PLACE OBJECT** generates a request for the **MODCA** scale-to-fill mapping option. Support for scale-to-fill depends on your printer and PSF level.
- Be sure your included object fits on the page.

DEFINE PATTERN Command

DEFINE PATTERN Command

With the **DEFINE PATTERN** command, you define an image, or overlay section, that you can place anywhere you wish as many times as you wish.

DEFINE — PATTERN Command



command word

DEFINE

section name Section names must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, --, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than eight characters.

section type Specify **PATTERN** if you intend to supply a coded pattern of toned and untoned pels that forms a printed image on the overlay.

pattern type There are two ways to code a pattern:

PELS To write a code made up of 1's and 0's that describes each pel individually.

ENCODED To write a code that describes *groups* of pels that are to be printed or not printed.

line coding **PELS** coding:

Code each line of pels separately and enclose each line in parentheses. Code the lines from left to right and from top to bottom.

For example, to define 2 toned pels (printed), 3 untoned pels (not printed), and then 4 toned pels, your **PELS** coding would look like this:

(110001111)

See page 121 for more information on pel coding.

ENCODED coding:

Write a numeric code that describes whether groups of pels should be toned or untoned. Write the number of untoned pels, leave a space, write the number of toned pels, leave a space, and write the number of untoned pels, and so on until you have defined the line. You must begin with untoned pels, but you need not end with untoned pels.

For example, to write a code to define 2 toned pels (printed), 3 untoned pels (not printed), and 4 toned pels, your **ENCODED** coding would look like this:

(0 2 3 4)

See page 121 for more information on encoded coding.

Note: Although the first pel in the example above is to be toned (printed), you must specify the number of initial untuned pels (in this case, 0).

Also, if you are coding a line that ends with untuned pels, you do not have to specify the number of trailing untuned pels. This means that you do not need to code 000011000 as (4 2 3); (4 2) is sufficient.

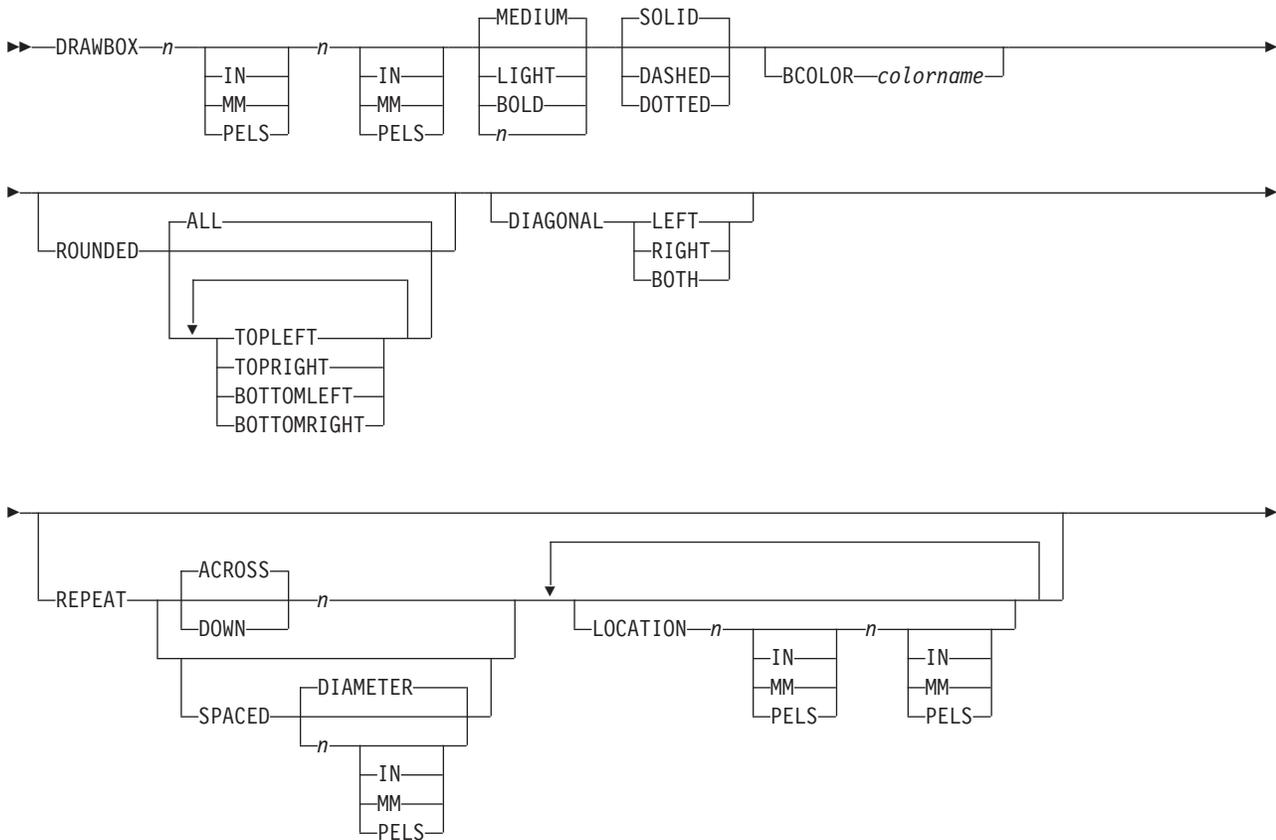
end marker Place an end marker (;) after the last coded line.

DRAWBOX Command

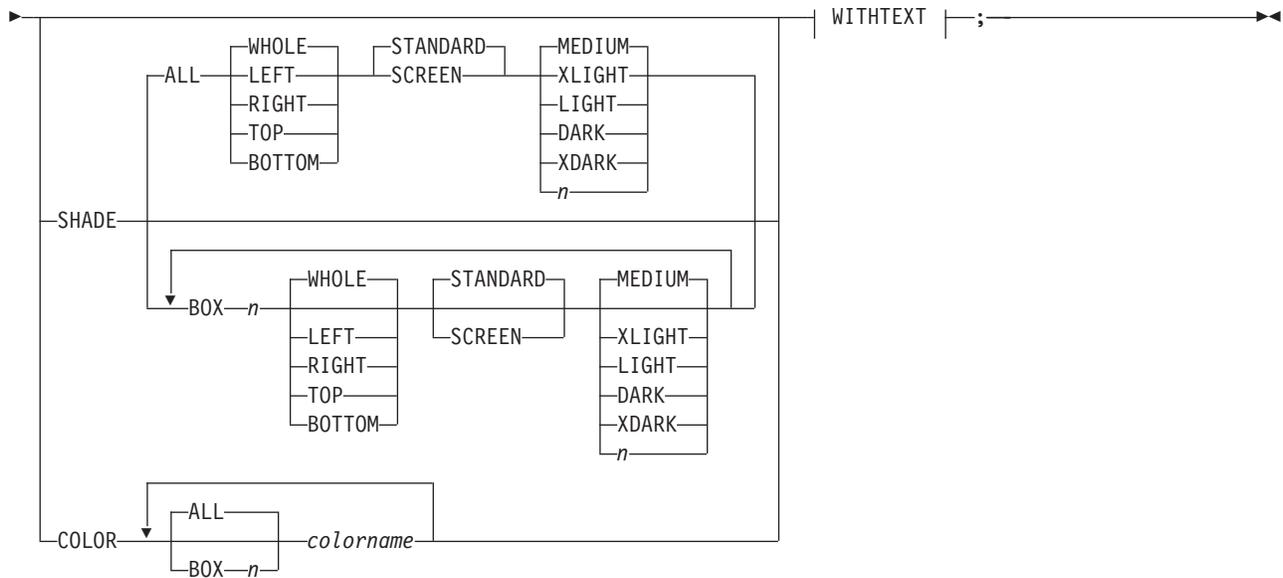
DRAWBOX Command

Use this command to draw boxes on your overlay. You can repeat, shade, and place text in any or all of the boxes you draw with this command.

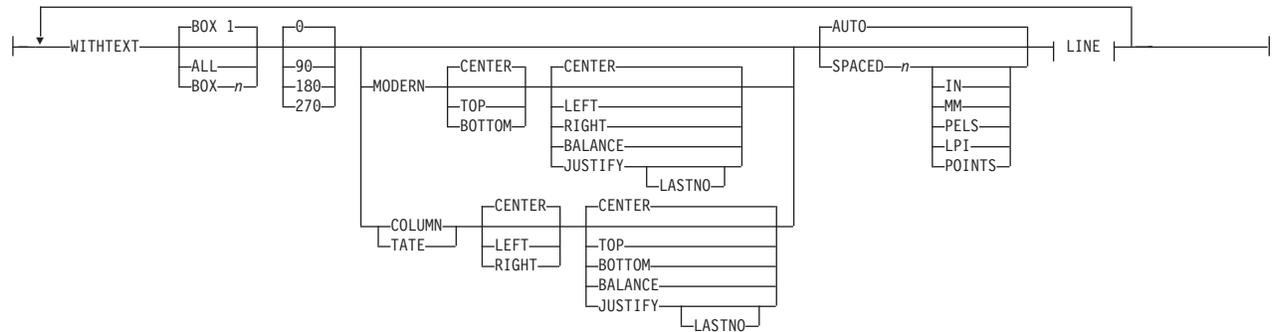
DRAWBOX Command



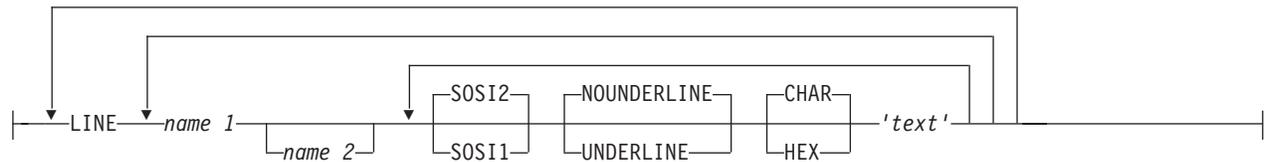
DRAWBOX Command



WITHTEXT:



LINE:



command word

DRAWBOX

box width

Enter a number (n) and a unit of measurement to specify how wide you want the box. Choose from:

n

- IN** Inches
- MM** Millimeters
- PELS** Pels

Note: If you do not enter a unit of measurement, the program uses the current value in the **SETUNITS** command. See "Using Default Options (**SETUNITS**)" on page 137.

DRAWBOX Command

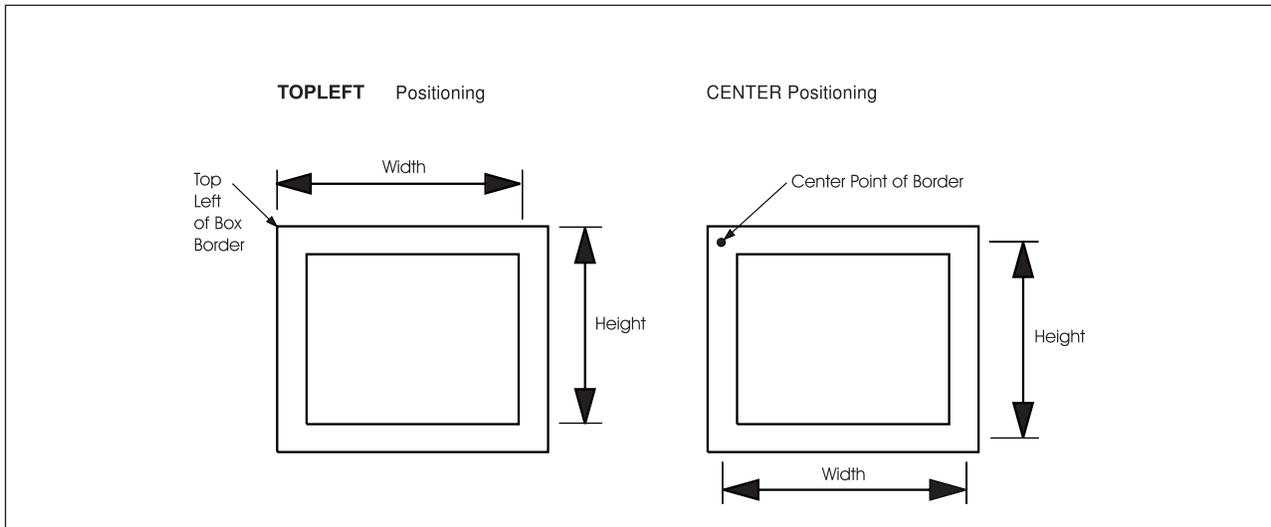


Figure 176. How to Measure the Width and Height of a Box

box height Enter a number (*n*) and a unit of measurement to specify the vertical dimension of the box. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

Note: If you do not enter a unit of measurement, the program uses the current value in the **SETUNITS** command.

Box width and height are measured depending on the positioning method being used, **TOPLEFT** or **CENTER**. For **TOPLEFT** positioning, the distance is measured from the top-left of the box border. For **CENTER** positioning, the distance is measured from the center point of the border at the top-left corner. This, however makes no difference to the overall size of the box.

border thickness

Specify the thickness of the border for the box. You can choose a ready-made border thickness, or customize a thickness by entering a number (*n*) in pels. Choose from:

LIGHT 2 pels wide.
MEDIUM 4 pels wide (default).
BOLD 6 pels wide.
n Thickness in pels, do not enter the *pels*.

Note: 0 thickness creates an invisible box, but text and shading can still be specified.

border type Specify the type of border you want surrounding the box. Choose from:

SOLID (Default)
DASHED
DOTTED

subcommand word

Specify **ROUNDED** if you want rounded corners on the box.

Notes:

1. The size of the rounded corners is the current value in the **SETUNITS** command.
2. If no **SETUNITS** command is specified before the **DRAWBOX** command, the corner rounding defaults to **MEDIUM**.

corner selection

Specify which corners you want rounded. Choose **ALL** or any combination of the other selections. Choose from:

ALL
TOPLEFT
TOPRIGHT
BOTTOMLEFT
BOTTOMRIGHT

subcommand word

Specify **DIAGONAL** if you want one or more diagonals in the box.

diagonal selection

Choose from:

LEFT Creates a diagonal from the top-left corner of the box to the bottom-right corner.
RIGHT Creates a diagonal from the top-right corner of the box to the bottom-left corner.
BOTH Creates both diagonals.

The diagonal has the same thickness and type as the box border.

end marker If you are drawing a single box without shading, color, or text, enter an end marker (;).

DRAWBOX COLOR Command

Use this command to add color to boxes.

BCOLOR Use **BCOLOR** to specify the color for all box borders in the **DRAWBOX**. If specified, **BCOLOR** must follow the box border type value.

| *colorname* Use *colorname* to specify the box border color. Multiple boxes done with one **DRAWBOX** command can have different border colors. The *colorname* parameter must occur after the **BCOLOR** subparameter.

If the color definition for box border rules has not been previously defined, the box rules will be done in the device default color. If the color definition for the box background color has not been previously defined, the color request is ignored.

| If **ROUNDED** and **COLOR** are both specified, only **ROUNDED** is honored. If **SHADE** and **COLOR** are both specified, only **COLOR** is honored. If **ROUNDED** is specified and the **ADVAFP** invocation option is used, the box shading is done with IM1 image.

DRAWBOX Command-Spaced Repetition

There are two ways to repeat boxes:

Spaced Repetition: Enter one set distance between boxes.

Location Repetition: Enter the location of each repeated box.

If the answer to both of the following questions is “yes”, use spaced repetition as described here. If the answer to either or both of the questions is “no”, see “**DRAWBOX** Command-Location Repetition” on page 211.

1. Are the box origins lined up either horizontally or vertically?
2. Is the spacing to be equal between the boxes?

DRAWBOX Command

subcommand word

Specify **REPEAT** if you want to draw more than one box of the same size, border thickness, border type, corner selection, and diagonal selection.

If spaced repetition is used, you may repeat any boxes beside or below the original.

direction

Choose from:

ACROSS The boxes are repeated to the right of the original box (default).

DOWN The boxes are repeated below the original box.

repetitions

Enter the number (*n*) of additional boxes.

spacing word **SPACED**

spacing value Using a number (*n*) and a unit of measurement, specify the space you want between the boxes. Choose from:

n

IN

MM Millimeters

PELS Pels

If you do not specify *spacing word* and *spacing value*, the program gives you boxes with 0 spacing. If you do not specify a unit of measurement, the program defaults to the spacing value from the last **SETUNITS** command.

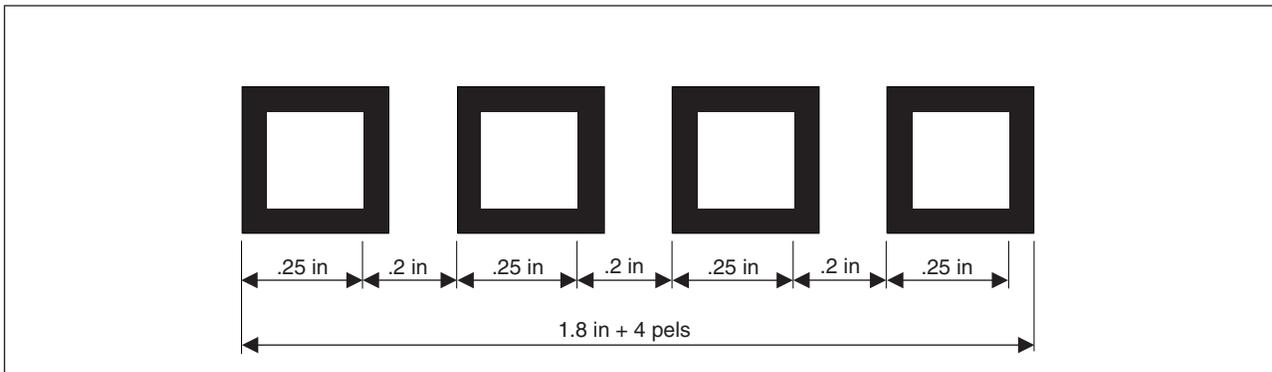


Figure 177. Spaced Boxes. These 0.25-inches-wide boxes with 4-pel borders are spaced 0.2 inches apart (not actual size).

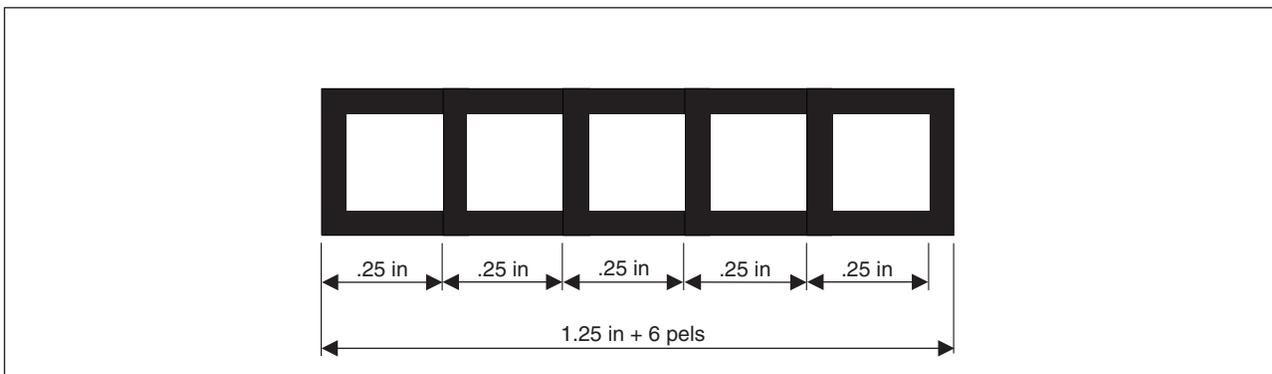


Figure 178. Boxes with 0 Spacing. These are 0.25-inches-wide boxes with 6-pel borders (not actual size).

DRAWBOX Command-Location Repetition

There are two ways to repeat boxes:

Location Repetition: Enter the location of each repeated box.

Spaced Repetition: Enter one set distance between boxes.

If the answer to either or both of the following questions is “no”, use location repetition as described here. If the answer to both of the following questions is “yes”, see “**DRAWBOX Command-Spaced Repetition**” on page 209.

1. Are the origins of the boxes in either horizontal or vertical straight lines?
2. Is the spacing to be equal between the boxes?

subcommand word

Specify **REPEAT** if you want to draw more than one box of the same size, border thickness, border type, corner selection, and diagonal selection.

location option Specify **LOCATION** to repeat the box anywhere within the overall dimensions of the overlay.

horizontal coordinate

Using a number (*n*) and a unit of measurement, specify the distance from the left edge of the overlay (or **GROUP** origin if the command is within a **GROUP** definition) to the point where you want the upper-left corner of the repeated box to begin. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the horizontal coordinate, the program defaults to the current value in the **SETUNITS** command.

vertical coordinate

Using a number (*n*) and a unit of measurement, specify the distance from the top of the overlay (or **GROUP** origin, if the command is part of a **GROUP** definition) to the point where you want to begin the upper-left corner of the box. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the vertical coordinate, the program defaults to the horizontal coordinate or current spacing value in the **SETUNITS** command.

DRAWBOX Command-Shading Boxes

shading option **SHADE**

box Choose from:

ALL Same shading in all of the boxes created by the current **DRAWBOX** command (default).

BOX *n* Shade a particular box in the overlay.

Note: Using spaced repetition, boxes are assigned numbers by their left to right or top to bottom order. If you used location repetition, box numbers are determined by the order in which they were defined.

DRAWBOX Command

shade area Specify the area of the box to be shaded. Choose from:

WHOLE (Default)
LEFT
RIGHT
TOP
BOTTOM

WHOLE refers to the entire interior of the box, including boxes with diagonals.

LEFT, **RIGHT**, **TOP**, and **BOTTOM** specify shading for the portions of the box marked off by the diagonals. For example, in a box with a **LEFT** diagonal, **SHADE LEFT** shades the area to the left of the diagonal. In this example, **SHADE BOTTOM** shades the same area.

Note: Specifying a particular shade area, for example, **SHADE LEFT**, on a box with no diagonals, shades the whole box.

If a box has a dotted or dashed border and the box portions are shaded using **LEFT**, **RIGHT**, **TOP**, and **BOTTOM**, the gaps between the dashes or dots in the diagonal are not shaded. If shaded with **WHOLE**, the gaps in the diagonal are shaded. The gaps in the border are never shaded.

shade pattern Choose from:

STANDARD Boxes shaded with the pattern shown in Figure 211 on page 326 (default).
SCREEN Boxes shaded with the pattern shown in Figure 212 on page 330.

shade type Specify how dark the shading should be for the boxes specified in the *box* entry.

To specify the amount of shading you want for the boxes:

- Enter one of the standard choices illustrated in Appendix G, “Shade Patterns and Types” on page 325 that describes the amount of shading you want. **MEDIUM** is the default.
- If you prefer shading that is lighter or darker than one of the standard choices, enter a number from 0 to 100 that matches the percentage of shading you want. See Appendix G, “Shade Patterns and Types” on page 325 for examples of all percentages.

| *colorname* Use *colorname* to specify the box background color. Multiple boxes done with one
| **DRAWBOX** command can have different background colors. The *colorname* parameter
| must occur after the **BCOLOR** subparameter.

|
| If the color definition for the box background color has not been previously defined, the
| color request is ignored.

|
| If **ROUNDED** and **COLOR** are both specified, only **ROUNDED** is honored. If **SHADE** and
| **COLOR** are both specified, only **COLOR** is honored. If **ROUNDED** is specified and the
| **ADVAFP** invocation option is used, the box shading is done with IM1 image.

end marker If the box or boxes you have defined in this command **do not** contain text, use an end
marker (;) to complete the **DRAWBOX** command.

DRAWBOX WITHTEXT Command

subcommand word

Specify **WITHTEXT** to place text in one or more of the boxes created by the current **DRAWBOX** command. See Chapter 4, “Adding Text” on page 65 for directions on how to define and to place text within a box.

box

Choose from:

ALL Places the same text in all of the boxes created by the current **DRAWBOX** command.

BOX 1 (Default).

BOX n Places text in a particular box defined in this command.

Note: If you use spaced repetition, boxes are assigned numbers by their left to right or top to bottom order. If you repeat boxes using location repetition, box numbers are determined by the order in which they were defined.

orientation

Specify the orientation of the text that is to be placed in the boxes. Choose from:

0 (Default)

90

180

270

Notes:

1. Remember that the text orientation you specify is relative to the overlay.
2. Before you specify orientation and format (the next entry), be sure that the font you select exists for that combination. See Appendix F, “Matching Fonts with Text Formatting” on page 323 to relate text orientation and format to font names, or ask your system programmer.

format

Select the format of the text you want placed in the specified box. Choose from:

MODERN Characters are printed from left to right, as the sentences appear on the pages of this manual (default).

COLUMN Characters are printed vertically from top to bottom and text strings are printed from left to right.

TATE Characters are printed vertically from top to bottom as in the **COLUMN** format, but text strings are printed right to left.

placement

Indicate the placement of text within the box. As you can see in the **DRAWBOX** syntax diagram, the text placement entries for **MODERN** (default), format are different from those for both the **COLUMN** and **TATE** formats.

Text placement is described in more detail in “Positioning and Spacing the Text” on page 76.

All text in boxes is aligned with reference to a text margin inside the box. Text margins are described in “Text Margins with **SETUNITS**” on page 144.

MODERN If you specify **MODERN**, choose an option from each of the following lists:

TOP

CENTER (Default)

BOTTOM

LEFT

CENTER

RIGHT

JUSTIFY

If you specify **JUSTIFY**, you have another option:

DRAWBOX Command

JUSTNO

If you specify **JUSTIFY LASTNO**, the last text string in the block is **LEFT** justified, but not **RIGHT** justified. See Figure 82 on page 87.

COLUMN or **TATE**

If you specify **COLUMN** or **TATE**, choose an option from each of the following lists:

TOP
CENTER (Default)
BOTTOM
LEFT
CENTER
RIGHT
JUSTIFY

If you specify **JUSTIFY**, you have another option:

JUSTNO

If you specify **JUSTIFY LASTNO**, the last text string in the block is **TOP** justified, but not **BOTTOM** justified.

line spacing If you are placing two or more text strings in the box, enter the type of line spacing you want to use:

AUTO Calculates the line spacing value based on the fonts used in the text strings (default).

SPACED Indicates that you intend to specify the spacing between text strings. To do this, enter a number (*n*) and a unit of measurement for the distance you want between the baselines of text. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch
POINTS	Points

Note: If you do not enter a unit of measurement, the program uses the current value in the **SETUNITS** command.

subcommand word

To begin defining a text string, you must enter the word:

LINE

You must enter the word **LINE** for each text string.

font names

Enter the names of the fonts you want to use for the text strings that follow. Specify either one or two fonts. One font is used for SBCS characters and the other is used for DBCS characters. The SBCS font specified is used for all SBCS characters in all subsequent text strings in the **LINE** subcommand until another SBCS font is designated. Similarly, the DBCS font specified remains in effect until another DBCS font is designated.

Note: The optional second font name is ignored when you specify the **NOSOSI** option in the **CONTROL** command.

If a text string contains any SBCS characters, you need to specify an SBCS font. If a text string contains any DBCS characters, you need to specify a DBCS font. If a single quoted text string contains both SBCS and DBCS characters, you need to specify a pair of fonts—one SBCS font and one DBCS font, the order is irrelevant.

DRAWBOX Command

If you use DBCS text and specify **SOSI1** mode, the SOSI delimiters appear as SBCS spaces. If you use **SOSI1** mode, you must specify an SBCS font in the **LINE** subcommand before you specify any text strings containing DBCS characters.

Each font name specified in a **LINE** subcommand must already be named in a previous **FONT** command.

To change fonts in a text string:

1. Complete the remaining **WITHTEXT** subcommand entries (underlining, text type, and text) for the text you want in the current font.
2. Enter the name of the new font or fonts you want to use and complete the remaining entries for that line of text.

SOSI mode This option defines the way **SOSI** delimiters are to be handled. This option is ignored if **NOSOSI** is specified in the **CONTROL** command.

SOSI1 A single-byte character space appears wherever SOSI delimiters occur.

When a SO delimiter follows single-byte text, the font used to determine the size of the character space is the same font used for the preceding single-byte text. When a SI delimiter is followed by single-byte text, the font used to determine the size of the character space is the same font used for the following single-byte text. If no single-byte text either precedes a SO delimiter or follows a SI delimiter, the last single-byte font specified is used to determine the size of the character spaces.

SOSI2 No space appears in the positions held by SOSI delimiters (default).

underlining Indicates whether you want the following text segment underlined. Blank spaces included as part of the text segment, are also underlined. Choose from:

NOUNDERLINE

Does not underline the following text segment (default).

UNDERLINE

Underlines the following text segment.

Notes:

1. Only text written in **MODERN** format can be underlined.
2. You must specify each text segment you want underlined.

text type Choose from:

CHAR The characters you type are the exact characters that are to be printed on the overlay (default).

HEX The text to follow is in hexadecimal form.

Notes:

1. Refer to page 112 for information on hex text.
2. For information about double-byte fonts, see Chapter 4, "Adding Text" on page 65.

text Enter the text you want placed in the boxes.

Each text entry that uses different characteristics (font, **UNDERLINE/NOUNDERLINE**, **CHAR/HEX**) must be enclosed in apostrophes as a unit.

Blanks in Balanced Text

If you specified **BALANCE** for the text placement, you should not include blanks between the apostrophes. However, if you want to include blanks in text that is to be balanced, you can do it by specifying multiple text segments: one segment for each character string (excluding blanks) and one segment for each string of blanks. For an example of how the balancing feature works, see "Blanks in Balanced Text" on page 96.

DRAWBOX Command

Symbolic Names

If there is text that you wish to use on several different overlays, you can give it a symbolic name. Then, each time you include the symbolic name in the **WITHTEXT** subcommand of the **DRAWBOX** command, the text represented by that symbolic name is printed on your overlay.

See “Symbolic Data Sets or Files” on page 114 for more information.

Notes:

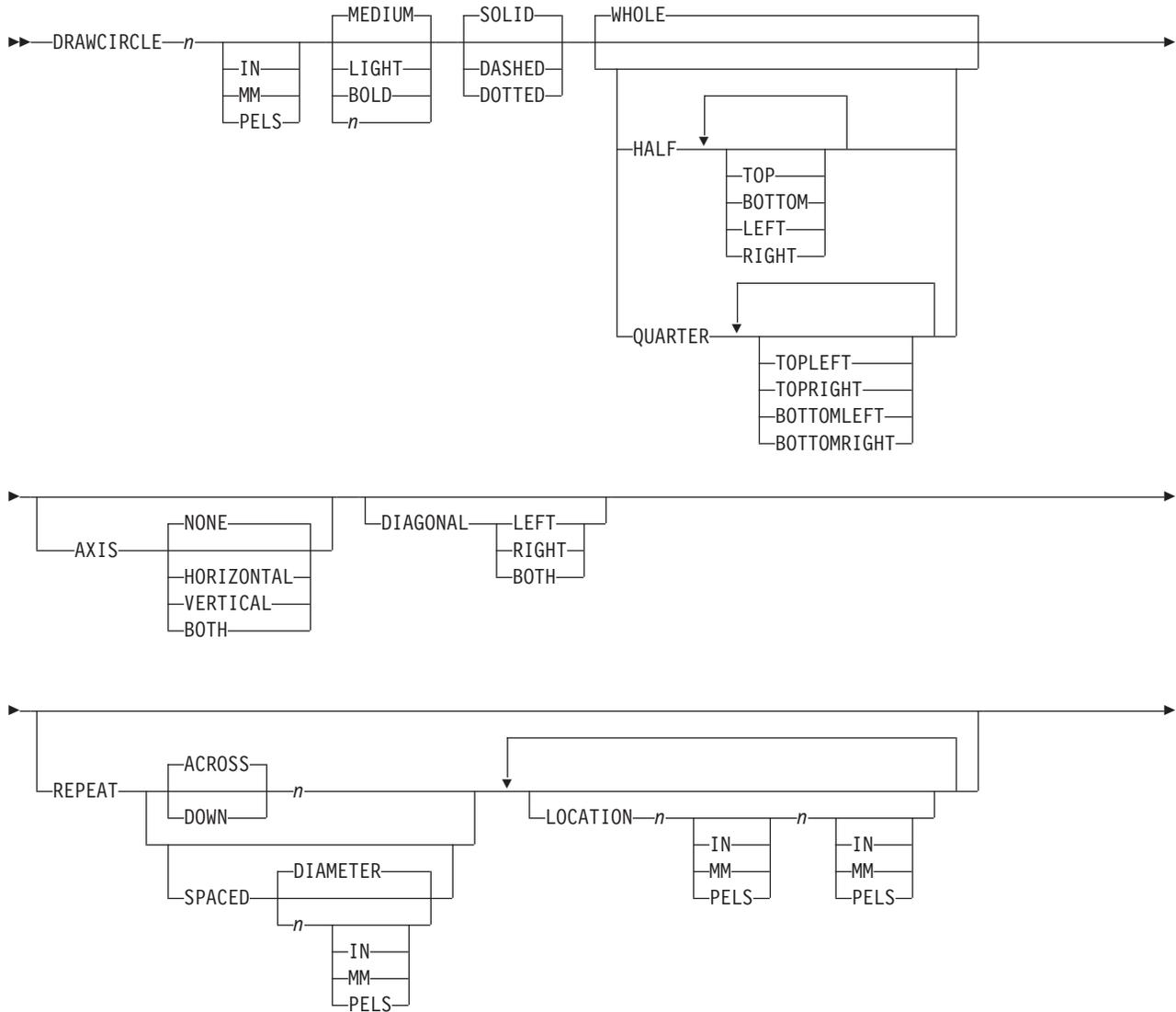
1. For single-byte fonts, precede a symbolic name with an ampersand (&) and follow it with a period.
2. For double-byte fonts, the symbolic name should begin with “&#”, and terminate with a period.

end marker Always end a command with an end marker (;).

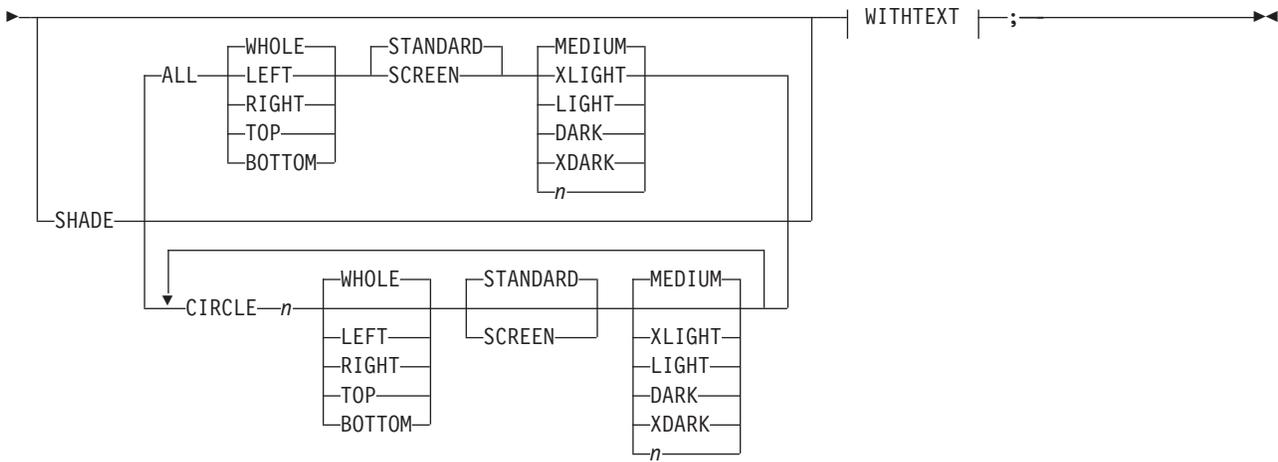
DRAWCIRCLE Command

Use this command to draw full or partial circles on your overlay. You can repeat, shade, and place text in any or all of the circles you draw with this command.

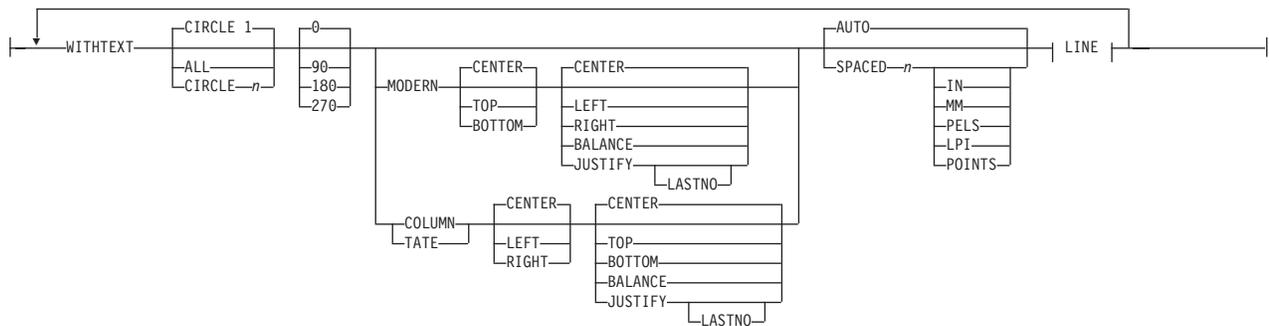
DRAWCIRCLE Command



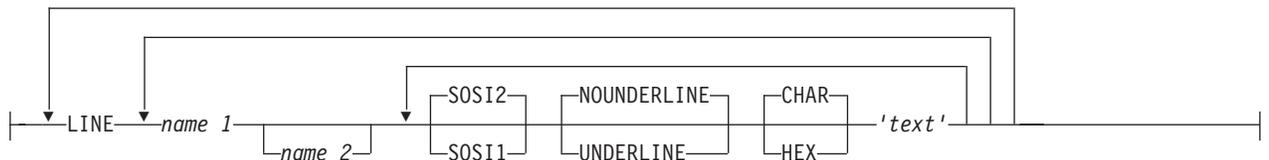
DRAWCIRCLE Command



WITHTEXT:



LINE:



command word

DRAWCIRCLE

circle radius Enter a number (*n*) and a unit of measurement to specify the radius of the circle. The radius is measured from the center of the circle, given in the last **POSITION** command, to the center line of the border. Choose from:

n

- IN** Inches
- MM** Millimeters
- PELS** Pels

Figure 179 on page 219 shows how the radius of a circle is measured.

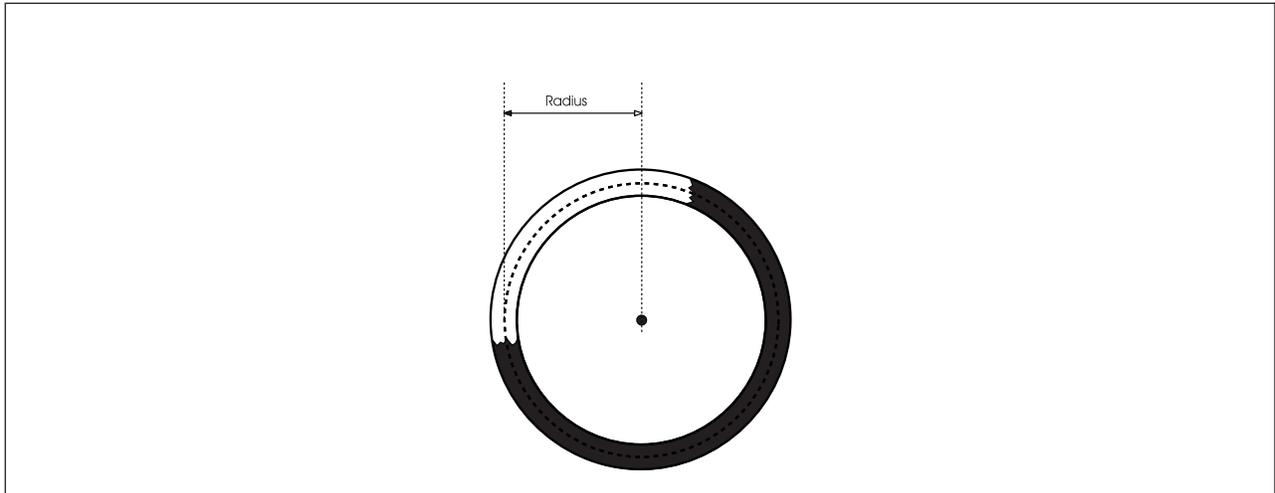


Figure 179. How to Measure the Radius of a Circle

Note: If you do not enter a unit of measurement, the program uses the default value from the last **SETUNITS** command. See “Using Default Options (**SETUNITS**)” on page 137.

border thickness

Specify the thickness of the border for the circle. The choices for border thickness are one of the following or a number (*n*) indicating the thickness in pels. Choose from:

LIGHT 2 pels wide.
MEDIUM 4 pels wide (default).
BOLD 6 pels wide.
n Thickness in pels, do not add the word “pels”.

Note: 0 thickness creates an invisible circle, but text and shading can still be specified.

border type

Choose from:

SOLID (Default)
DASHED
DOTTED

circle portion

Specify whether you want a **WHOLE** or partial circle drawn. Choose from:

WHOLE A whole circle is drawn (default).
HALF Choose from:
 TOP
 BOTTOM
 LEFT
 RIGHT
QUARTER Choose from:
 TOP
 BOTTOM
 LEFT
 RIGHT

Note: It is possible to specify more than one circle portion. You can combine $\frac{1}{4}$ -circles and $\frac{1}{2}$ -circles to create full circles or $\frac{3}{4}$ -circles. See page 56 for examples.

end marker

If you are drawing a single circle without diagonals, shading, or text, enter an end marker (;).

DRAWCIRCLE Command

subcommand word

Enter one of the following words to alter the way dots and dashes are placed around the circle:

AXIS or **AXES**

Note: The choice has no effect on the result. This subcommand only has noticeable effect when the thickness of the line is of the same order as the radius of the circle. If **AXIS** or **AXES** is not used, the default, **NONE**, is applied.

axis selection Choose from:

NONE (Default)

HORIZONTAL

VERTICAL

BOTH

The axis selections are illustrated in Figure 180.

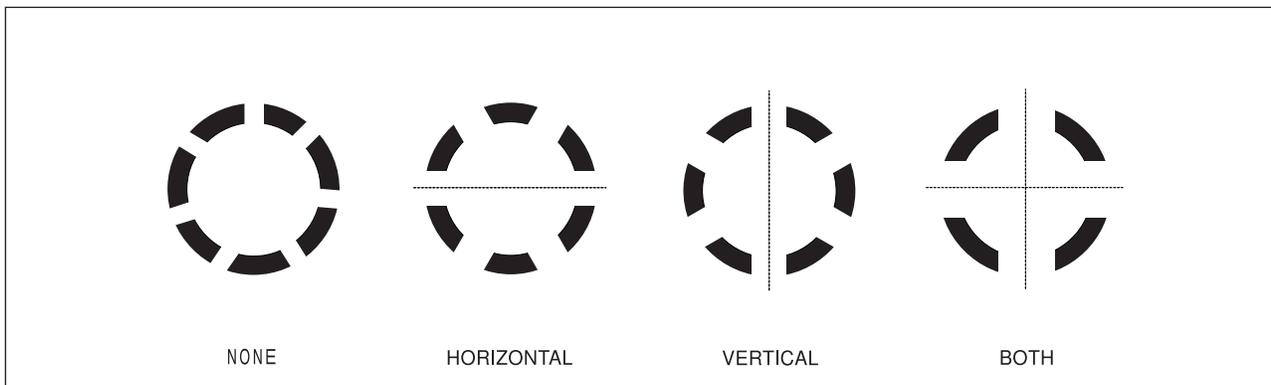


Figure 180. Effects produced using the **AXIS (AXES)** subcommand

subcommand word

Enter **DIAGONAL** to place diagonals in the circle.

Note: Only **WHOLE** circles can have diagonals placed inside them.

diagonal selection

Choose from:

LEFT Creates a diagonal from the top-left to the bottom-right of the circle.

RIGHT Creates a diagonal from the top-right to the bottom-left of the circle.

BOTH Creates both diagonals.

Note: There is no default value for diagonal selection.

end marker

If you are drawing a single circle without shading or text, enter an end marker (;).

DRAWCIRCLE Command-Spaced Repetition

There are two ways to repeat circles:

Spaced Repetition: Enter one set distance between circles.

Location Repetition: Enter the location of each repeated circle.

If the answer to both of the following questions is “yes”, use spaced repetition as described here. If the answer to either or both of the questions is “no”, see “**DRAWCIRCLE** Command-Location Repetition” on page 223.

1. Are the circle centers lined up either horizontally or vertically?
2. Is the spacing to be equal between the circles?

subcommand word

Specify **REPEAT** if you want to draw more than one circle of the same size, border thickness, border type, circle portion, and diagonal selection.

If spaced repetition is used, you may repeat any circles beside or below the original.

direction

Choose from:

ACROSS The circles are repeated to the right of the original circle (default).
DOWN The circles are repeated below the original circle.

repetitions

Enter the number (*n*) of additional circles.

spacing word

SPACED

spacing value

Using a number (*n*) and a unit of measurement, specify the space you want to leave between the centers of the circles.

The spacing between circles is measured from the center of one circle to the center of the next. See Figure 181 on page 222. Choose from:

DIAMETER (Default)

n

IN Inches
MM Millimeters
PELS Pels

If no value is specified, the unit of measurement defaults to the **SETUNITS** value.

Notes:

1. The default spacing (**DIAMETER**), causes the circle centers to be spaced one diameter apart, which means that the diameters are just touching one another. See Figure 182 on page 222.
2. If you do not specify *spacing word* and *spacing value*, the program gives you circles with **DIAMETER** spacing.
3. If you specify a spacing value of 0, the space between the centers of the circles is 0. In other words, the repeated circles are not visible.

DRAWCIRCLE Command

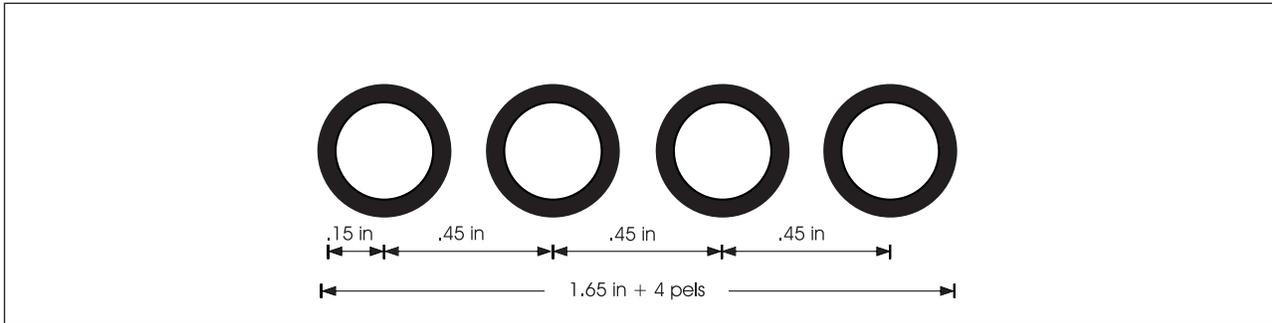


Figure 181. Spaced Circles. (Not actual size.)

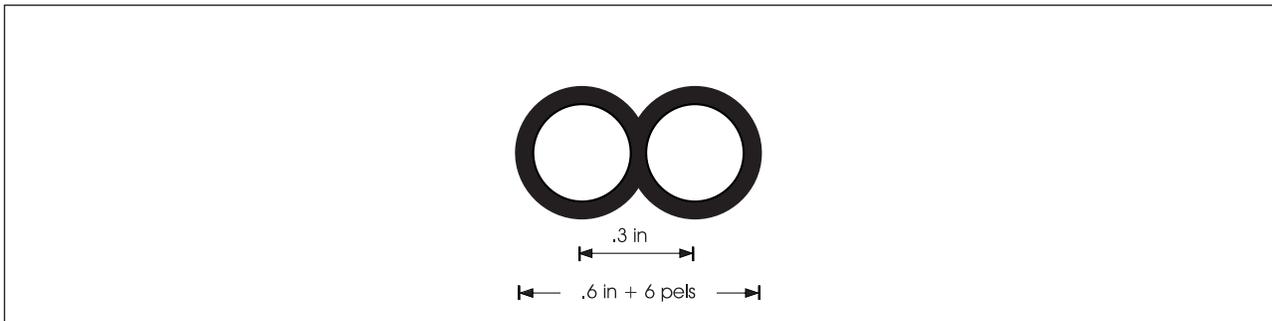


Figure 182. Circles with **DIAMETER** Spacing. These are circles with radius 0.15 inches and 6 pel borders (not actual size).

DRAWCIRCLE Command-Location Repetition

There are two ways to repeat circles:

Location Repetition: Enter the location of each repeated circle.

Spaced Repetition: Enter one set distance between circles.

If the answer to either or both of the following questions is “no”, use location repetition, as described here. If the answer to both of the following questions is “yes”, see “**DRAWCIRCLE Command-Spaced Repetition**” on page 221.

Notes:

1. Are the centers of the circles in either horizontal or vertical straight lines?
2. Is the spacing to be equal between the circles?

subcommand word

Specify **REPEAT** if you want to draw more than one circle of the same size, border thickness, border type, circle portion, and diagonal selection.

location option Specify **LOCATION** to repeat the circle anywhere within the overall dimensions of the overlay.

horizontal coordinate

Using a number (*n*) and a unit of measurement, specify the distance from the left edge of the overlay (or **GROUP** origin, if the command is within a **GROUP** definition) to the point where you want the center of the repeated circle. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the horizontal coordinate, the program defaults to the current value in the **SETUNITS** command.

vertical coordinate

Using a number (*n*) and a unit of measurement, specify the distance from the top of the overlay (or **GROUP** origin if the command is part of a **GROUP** definition) to the point where you want the center of the repeated circle. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the vertical coordinate, the program defaults to the horizontal coordinate or the current value in the **SETUNITS** command.

DRAWCIRCLE Command

DRAWCIRCLE Command-Shading Circles

shading option **SHADE**

circle Choose from:

ALL The same shading in all of the circles created by the current **DRAWCIRCLE** command (default).

CIRCLE *n* Shade a particular circle in the command.

Note: Using spaced repetition, circles are assigned numbers by their left to right or top to bottom order. If you used location repetition, circle numbers are determined by the order in which they were defined.

shade area Specify the area of the whole or partial circle to be shaded. Choose from:

WHOLE (Default)

LEFT

RIGHT

TOP

BOTTOM

No matter what option you pick for a partial circle, all of the area of the partial circle will be shaded.

LEFT, **RIGHT**, **TOP**, and **BOTTOM** specify shading for the portions of the circle marked off by the diagonals. For example, in a circle with a **LEFT** diagonal, **SHADE LEFT** shades the area to the left of the diagonal. In this example, **SHADE BOTTOM** shades the same area.

Note: Specifying a particular shade area, for example, **SHADE LEFT**, on a circle with no diagonals, shades the whole circle.

If a circle has a dotted or dashed border and the box portions are shaded using **LEFT**, **RIGHT**, **TOP**, and **BOTTOM**, the gaps between the dashes or dots in the diagonal are not shaded. If shaded with **WHOLE**, the gaps in the diagonal are shaded. The gaps in the border are never shaded.

shade pattern Choose from:

STANDARD All circles shaded with the pattern shown in Figure 211 on page 326 (default).

SCREEN All circles shaded with the pattern shown in Figure 212 on page 330.

shade type Specify how dark the shading should be for the circle(s) specified in the *circle* entry.

To specify the amount of shading you want for the circle(s):

- Enter one of the standard choices illustrated in Appendix G, “Shade Patterns and Types” on page 325 that describes the amount of shading you want. **MEDIUM** is the default.
- If you prefer shading that is lighter or darker than one of the standard choices, enter a number from 0 to 100 that matches the percentage of shading you want. See Appendix G, “Shade Patterns and Types” on page 325 for examples of all percentages.

end marker If the circle or circles you have defined in this command do not contain text, use an end marker (;) to complete the **DRAWCIRCLE** command.

DRAWCIRCLE WITHTEXT Command*subcommand word*

Specify **WITHTEXT** to place text in one or more of the circles created by the current **DRAWCIRCLE** command. See “Adding Text to Circles (**DRAWCIRCLE WITHTEXT**)” on page 106 for directions on how to define and to place text within a circle.

circle

Choose from:

ALL Places the same text in all the circles created by the current **DRAWCIRCLE** command.

CIRCLE 1 (Default)

CIRCLE *n* Places text in a particular circle for this command.

Note: If you use spaced repetition, circles are assigned numbers by their left to right or top to bottom order. If you repeat circles using location repetition, box numbers are determined by the order in which they were defined.

orientation

Specify the orientation of the text that is to be placed in the circle(s). Choose from:

0 (Default)

90

180

270

Notes:

1. Remember that the text orientation you specify is relative to the overlay.
2. Before you specify orientation and format (the next entry), be sure that the font you select exists for that combination. See Appendix F, “Matching Fonts with Text Formatting” on page 323 to relate text orientation and format to font names, or ask your system programmer.

format

Select the format of the text you want placed in the specified circle. Choose from:

MODERN Characters are printed from left to right, as the sentences appear on the pages of this manual (default).

COLUMN Characters are printed vertically from top to bottom, and text strings are printed from left to right.

TATE Characters are printed vertically from top to bottom as in the **COLUMN** format, but text strings are printed right to left.

placement

Indicate the placement of text within the circle. As you can see in the **DRAWCIRCLE** syntax diagram, the text placement entries for **MODERN** (default), format are different from those for both the **COLUMN** and **TATE** formats.

Text placement is described in more detail in “Adding Text to Circles (**DRAWCIRCLE WITHTEXT**)” on page 106 and “Positioning and Spacing the Text” on page 76.

All text in circles is oriented with reference to a text margin inside the circle. Text margins are described in “Text Margins with **SETUNITS**” on page 144.

MODERN If you specify **MODERN**, choose an option from each of the following lists:

TOP

CENTER (Default)

BOTTOM

LEFT

CENTER

RIGHT

BALANCE

JUSTIFY

If you specify **JUSTIFY**, you have another option:

DRAWCIRCLE Command

JUSTNO

If you specify **JUSTIFY LASTNO**, the last text string in the block is **LEFT** justified, but not **RIGHT** justified. See Figure 83 on page 88.

COLUMN or TATE

If you specify **COLUMN** or **TATE**, choose an option from each of the following lists:

TOP

CENTER (Default)

BOTTOM

LEFT

CENTER

RIGHT

BALANCE

JUSTIFY

If you specify **JUSTIFY**, you have another option:

JUSTNO

If you specify **JUSTIFY LASTNO**, the last text string in the block is **TOP** justified, but not **BOTTOM** justified.

line spacing If there are two or more text strings you are placing in the circle, enter the type of line spacing you want to use:

AUTO

Applies the standard line spacing for the font you are using. For lines containing a number of fonts, calculations are performed using the largest font ascender and descender in each line (default).

SPACED

Indicates that you intend to specify the spacing between text strings. To do this, enter a number (*n*) and a unit of measurement for the distance you want between the baselines of text. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch
POINTS	Points

Note: If you do not specify a unit of measurement, the program defaults to the current value in the **SETUNITS** command.

subcommand word

To begin defining a text string, you must enter the word:

LINE

You must enter the word **LINE** for each text string.

font names

Enter the names of the fonts you want to use for the text strings that follow. Specify either one or two fonts. One font is used for SBCS characters and the other is used for DBCS characters. The SBCS font specified is used for all SBCS characters in all subsequent text strings in the **LINE** subcommand until another SBCS font is designated. Similarly, the DBCS font specified remains in effect until another DBCS font is designated.

Note: The optional second font name is ignored when you specify the **NOSOSI** option in the **CONTROL** command.

If a text string contains any SBCS characters, you need to specify an SBCS font. If a text string contains any DBCS characters, you need to specify a DBCS font. If a single quoted

DRAWCIRCLE Command

text string contains both SBCS and DBCS characters, you need to specify a pair of fonts—ONE SBCS font and one DBCS font, the order is irrelevant.

If you use DBCS text and specify **SOSI1** mode, the SO and SI characters appear as SBCS spaces. If you use SOSI1 mode, you must specify an SBCS font in the **LINE** subcommand before you specify and text strings containing DBCS characters.

Each font name specified in the **LINE** subcommand must already be named in a previous **FONT** command.

To change fonts in a text string:

1. Complete the remaining **WITHTEXT** subcommand entries (underlining, text type, and text) for the text you want in the current font; and
2. Enter the name of the new font or fonts you want to use, and complete the remaining entries for that line of text.

SOSI mode This option defines the way SOSI delimiters are to be handled. This option is ignored if **NOSOSI** is specified in the **CONTROL** command.

SOSI1 A single-byte character space appears wherever SOSI delimiters occur.

When a SO delimiter follows single-byte text, the font used to determine the size of the character space is the same font used for the preceding single-byte text.

When a SI delimiter is followed by single-byte text, the font used to determine the size of the character space is the same font used for the following single-byte text. If no single-byte text either precedes a SO delimiter or follows a SI delimiter, the last single-byte font specified is used to determine the size the character spaces.

SOSI2 No space appears in the positions held by SOSI delimiters (default).

underlining Indicates whether you want the following text segment underlined. Blank spaces included as part of the text segment are also underlined. Choose from:

NOUNDERLINE

Does not underline the following text segment (default).

UNDERLINE Underlines the following text segment.

Notes:

1. Only text written in **MODERN** format can be underlined.
2. You must specify each text segment you want underlined. See page 227.

text type Specify text type as follows:

CHAR The characters you type are the exact characters that are to be printed on the overlay (default).

HEX The text to follow is in hexadecimal form.

Notes:

1. Refer to page 112 for information on hex text.
2. For information about double-byte fonts, see “Symbolic Data Sets and Symbolic Files Containing Double-Byte Characters” on page 320.

text Enter the text you want placed in the overlay. Each part of a line that uses different characteristics (**FONT**, **UNDERLINE/NOUNDERLINE**, **CHAR/HEX**) must be enclosed within apostrophes as a unit.

Blanks in Balanced Text

If you specified **BALANCE** for the text placement, you should not include blanks between the apostrophes. However, if you want to include blanks in text that is to be balanced, you

DRAWCIRCLE Command

can do it by specifying multiple text entries; one entry for each character string (excluding blanks) and one entry for each string of blanks. For an example of how this balancing feature works, see “Blanks in Balanced Text” on page 96.

Symbolic names

If there is text that you want to use on several different overlays, you can give it a symbolic name. Then, each time you include the symbolic name in the **WITHTEXT** subcommand of the **DRAWCIRCLE** command, you get the text it represents on your overlay.

See “Symbolic Data Sets or Files” on page 114 for more information.

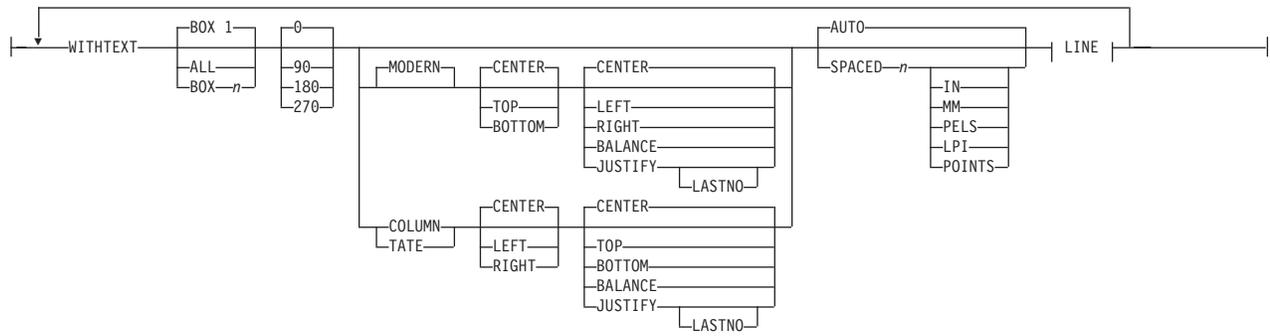
Notes:

1. For single-byte fonts, precede a symbolic name with an ampersand (&) and follow it with a period.
2. For double-byte fonts, the symbolic name should begin with “&#”, and terminate with a period.

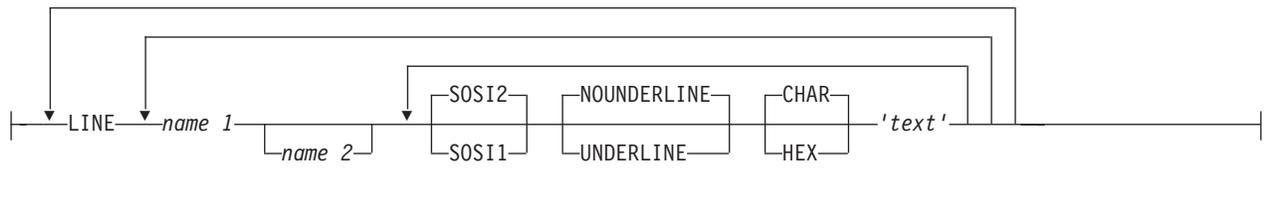
end marker Always end a command with an end marker (;).

DRAWGRAPHIC Command

WITHTEXT:

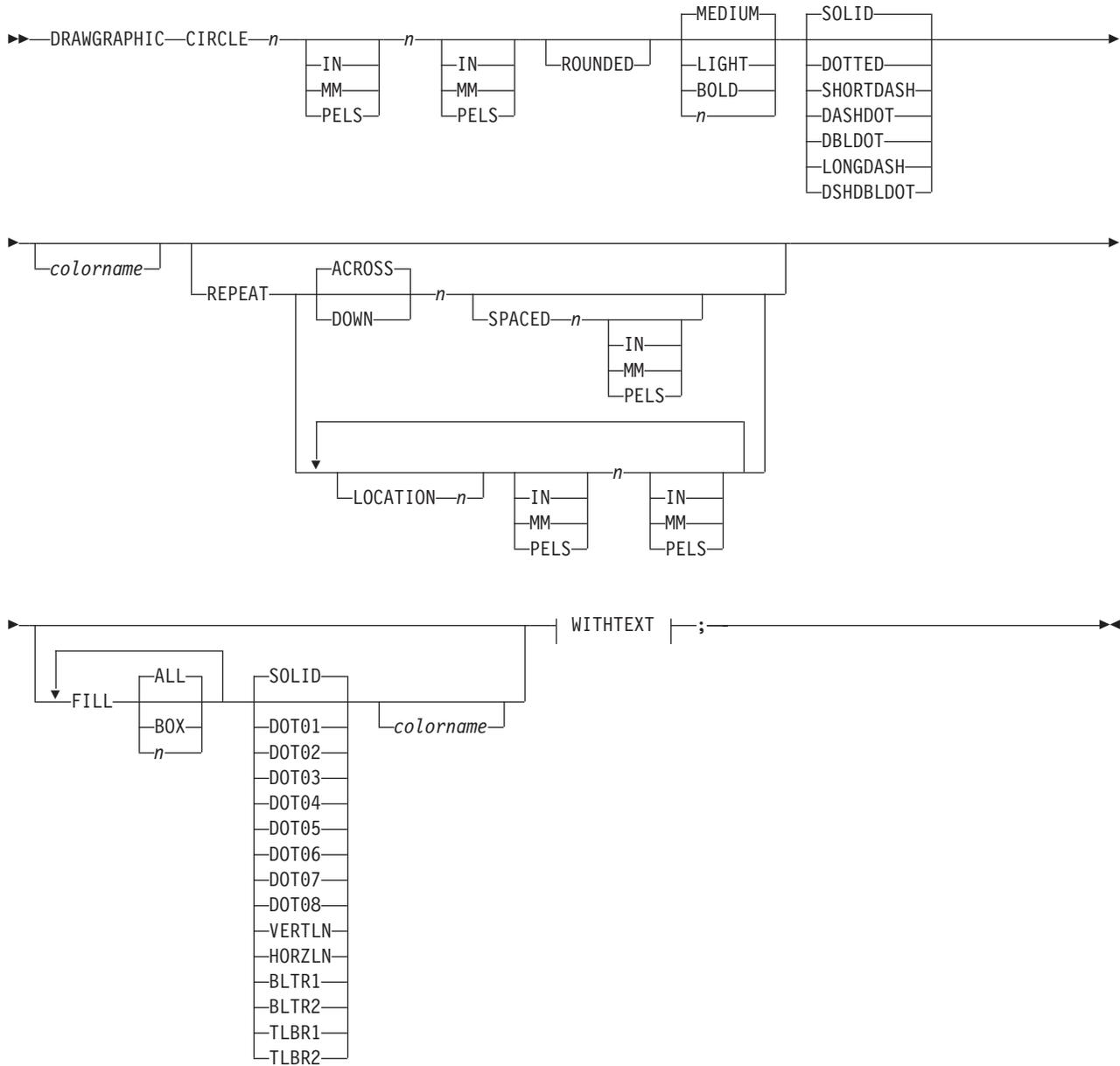


LINE:



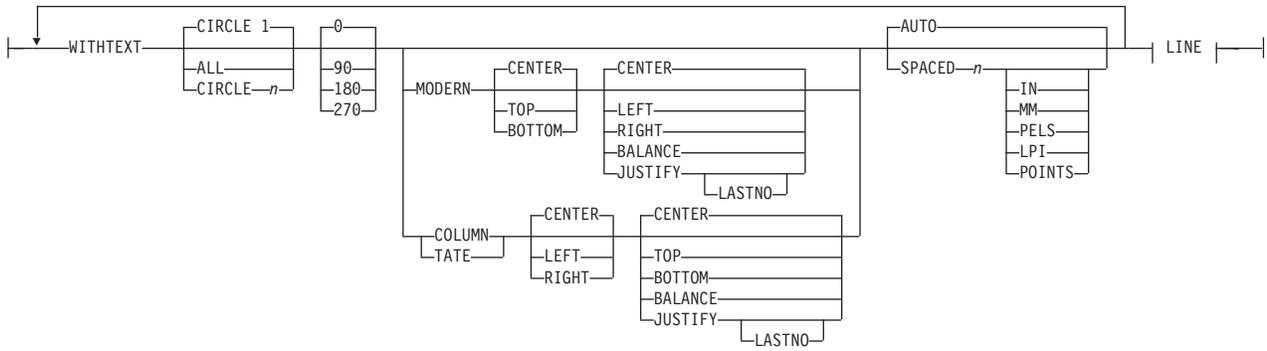
DRAWGRAPHIC CIRCLE Command

DRAWGRAPHIC — CIRCLE Command

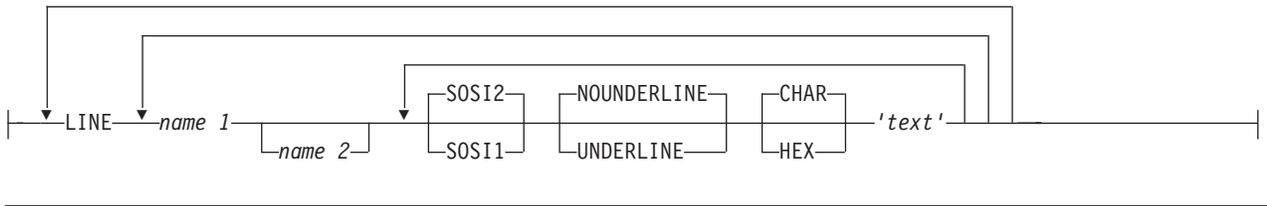


WITHTEXT:

DRAWGRAPHIC Command

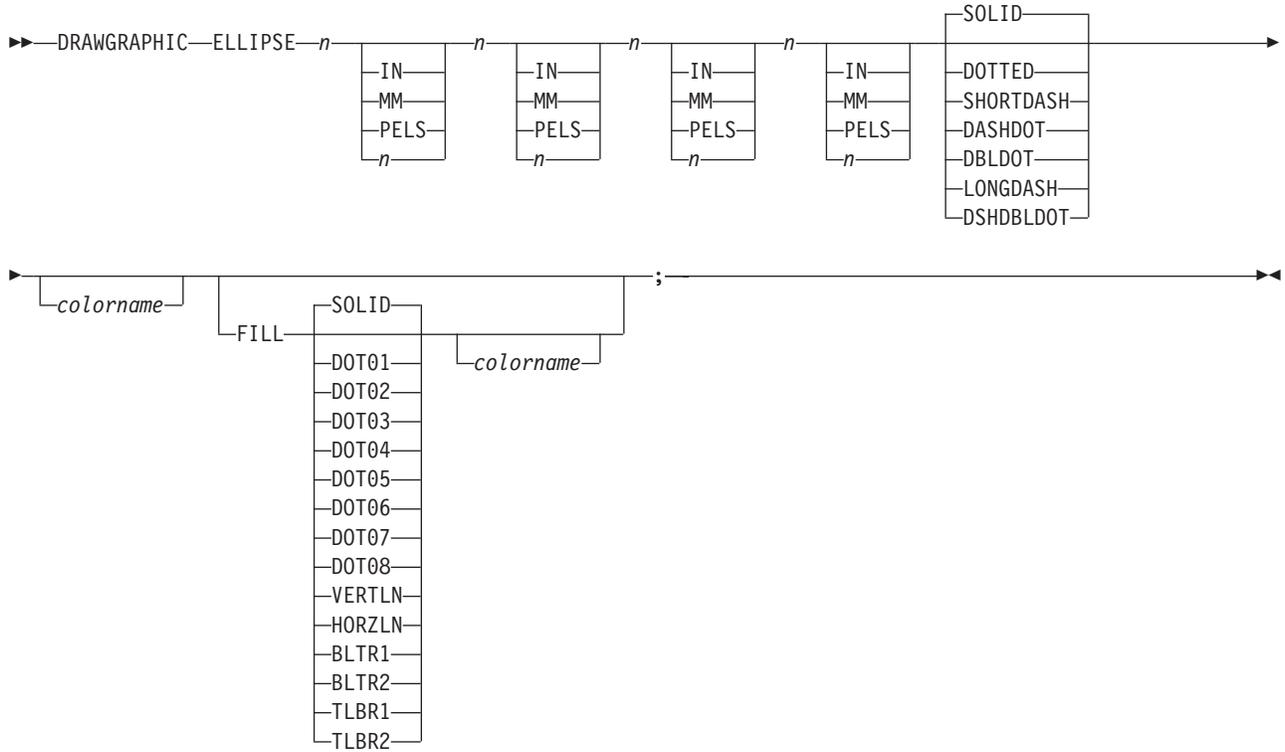


LINE:



DRAWGRAPHIC ELLIPSE Command

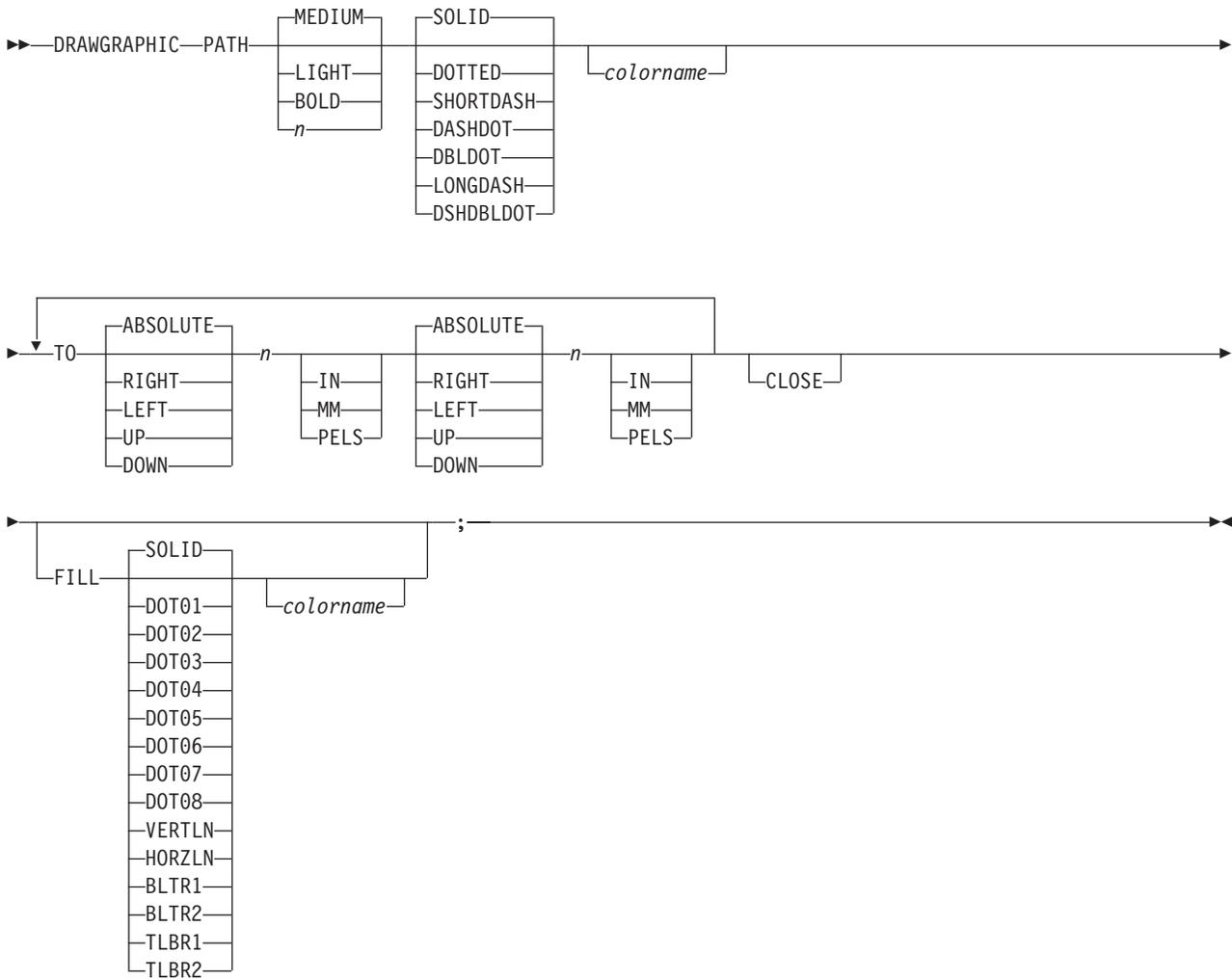
DRAWGRAPHIC — ELLIPSE Command



DRAWGRAPHIC Command

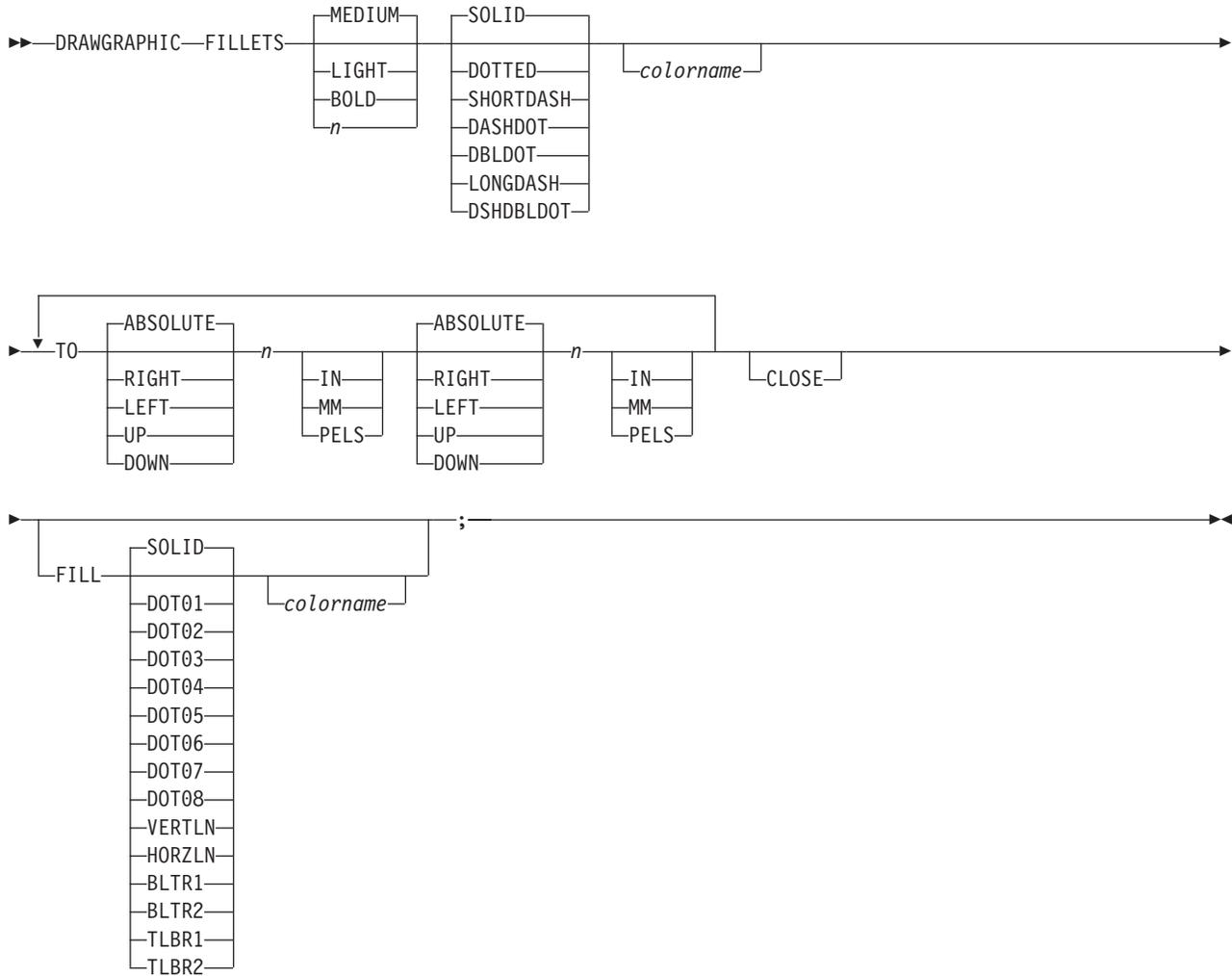
DRAWGRAPHIC PATH Command

DRAWGRAPHIC Command — PATH



DRAWGRAPHIC FILLETS Command

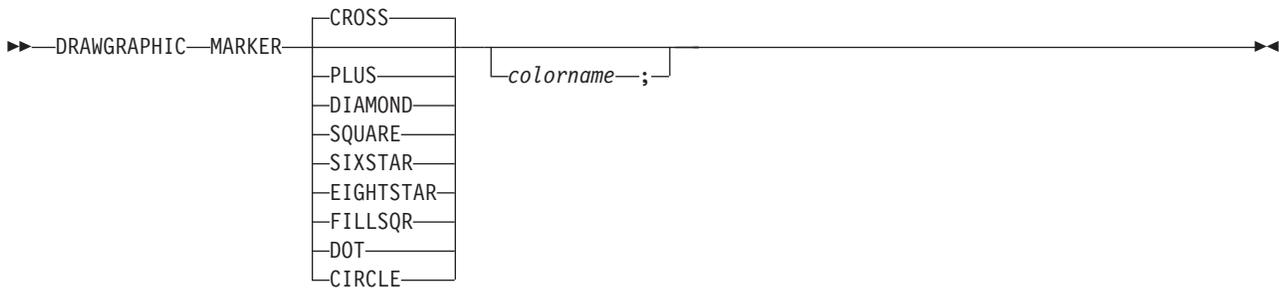
DRAWGRAPHIC — FILLETS Command



DRAWGRAPHIC Command

DRAWGRAPHIC MARKER Command

DRAWGRAPHIC — MARKER Command



type Select the type of object that you want to draw. Valid types are:

- CIRCLE** Draws a circle with the specified radius, using the specified or default linewidth and linetype. The circle will be filled if you specify a fill pattern.
- Circles can be repeated with **SPACED** or **LOCATION** parameters similar to the **DRAWCIRCLE** command. The position of each circle will be its center.
- ELLIPSE** Draws an ellipse, using the specified *x* and *y* coordinates for the end of the major and minor axes. The ellipse will be drawn with the specified or default linewidth and linetype, and will be filled if you specify a fill pattern. See Figure 183 on page 237 for a diagram.

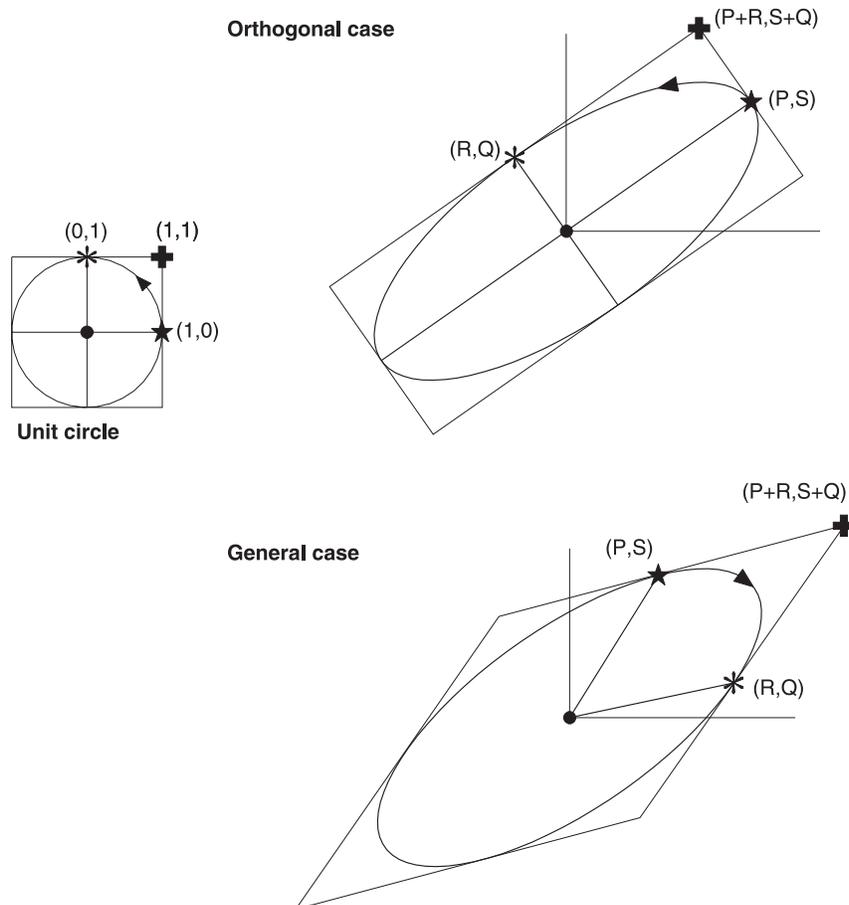


Figure 183. Ellipse parameters

- The origin of the ellipse will be at the current overlay position.
- BOX** Draws a box, with either all square or all rounded corners, for the specified dimensions. The box will be drawn with the specified or default linewidth and linetype, and will be filled if you specify a fill pattern. The default is square corners. If rounded corners are specified, the length of the “round” part of the box is determined by the **CORNERLENGTH** parameter on the **SETUNITS** command, similar to **DRAWBOX**.
- Boxes can be repeated with **SPACED** or **LOCATION** parameters similar to the **DRAWBOX** command. The position of each box will be its upper left-hand corner.
- PATH** Draws a series of straight lines, at the specified X,Y positions from the page origin. The thickness and type of line is determined from the *linewidth* and *linetype* parameters or their defaults. If the lines complete a closed figure, the figure will be filled if you specify a fill pattern.
- FILLETS** Draws curves that are created by joining the points with conceptual straight lines, and then drawing a curve that is tangential to the first line at its start point, the last line at its end point, and intermediate lines at their center points. The thickness and type of line is determined from the *linewidth* and *linetype* parameters or their defaults. See Figure 184 on page 238 for an example.

DRAWGRAPHIC Command

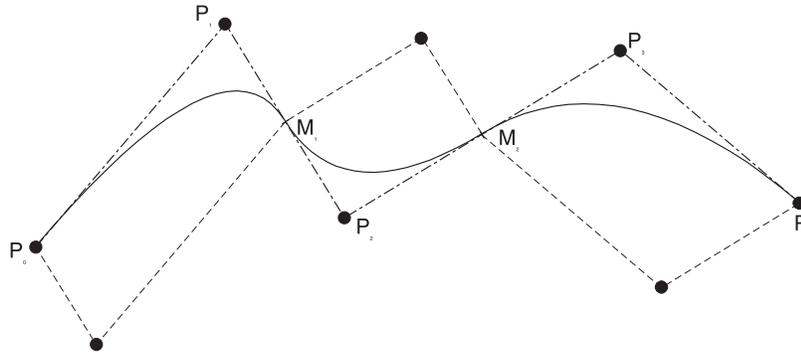


Figure 184. Fillets

MARKER Symbols that are used to indicate a position. The particular symbol that is drawn is determined by the *marker symbol* parameter. A marker will be drawn at the current overlay position.

linewidth Used to control the width of straight or curved lines only within this graphic object. The value of the line width parameter is an integer value representing the number of one hundredths (.01) of an inch of width. For example, 2 = .02 of an inch.

LIGHT 1 standard line width
MEDIUM 2 standard line widths (default)
BOLD 3 standard line widths
n *n* standard line widths

linetype Specifies what type of line to draw. The options are:

DOTTED dotted line
SHORTDASH short dashed line
DASHDOT dash-dotline
DBLDOT double dotted line
LONGDASH long dashed line
DSHDBLDOT dash double-dot line
SOLID solid line

colorname Specifies the color to be used. The *colorname* refers to a color definition previously created with the **DEFINE COLOR** command.

marker symbol Specifies which of the following symbols should be used to draw markers:

CROSS two crossed lines (X)
PLUS plus sign
DIAMOND hollow diamond
SQUARE hollow square
SIXSTAR 6-pointed star
EIGHTSTAR 8-pointed star
FILLDMND filled diamond
FILLSQR filled square
DOT solid dot
CIRCLE hollow dot

The default is **CROSS** if the marker symbol is not specified.

fillpattern Controls which pattern is used to fill objects. The patterns are shown in Figure 185 on page 240 and can be any of the following. The default is solid fill.

DOT01 — DOT08
dotted patterns of decreasing density

DRAWGRAPHIC Command

VERTLN	vertical lines
HORZLN	horizontal lines
BLTR1 & BLTR2	diagonal lines from bottom-left to top-right
TLBR1 & TLBR2	diagonal lines from top-left to bottom-right
<u>SOLID</u>	solid fill

DRAWGRAPHIC Command

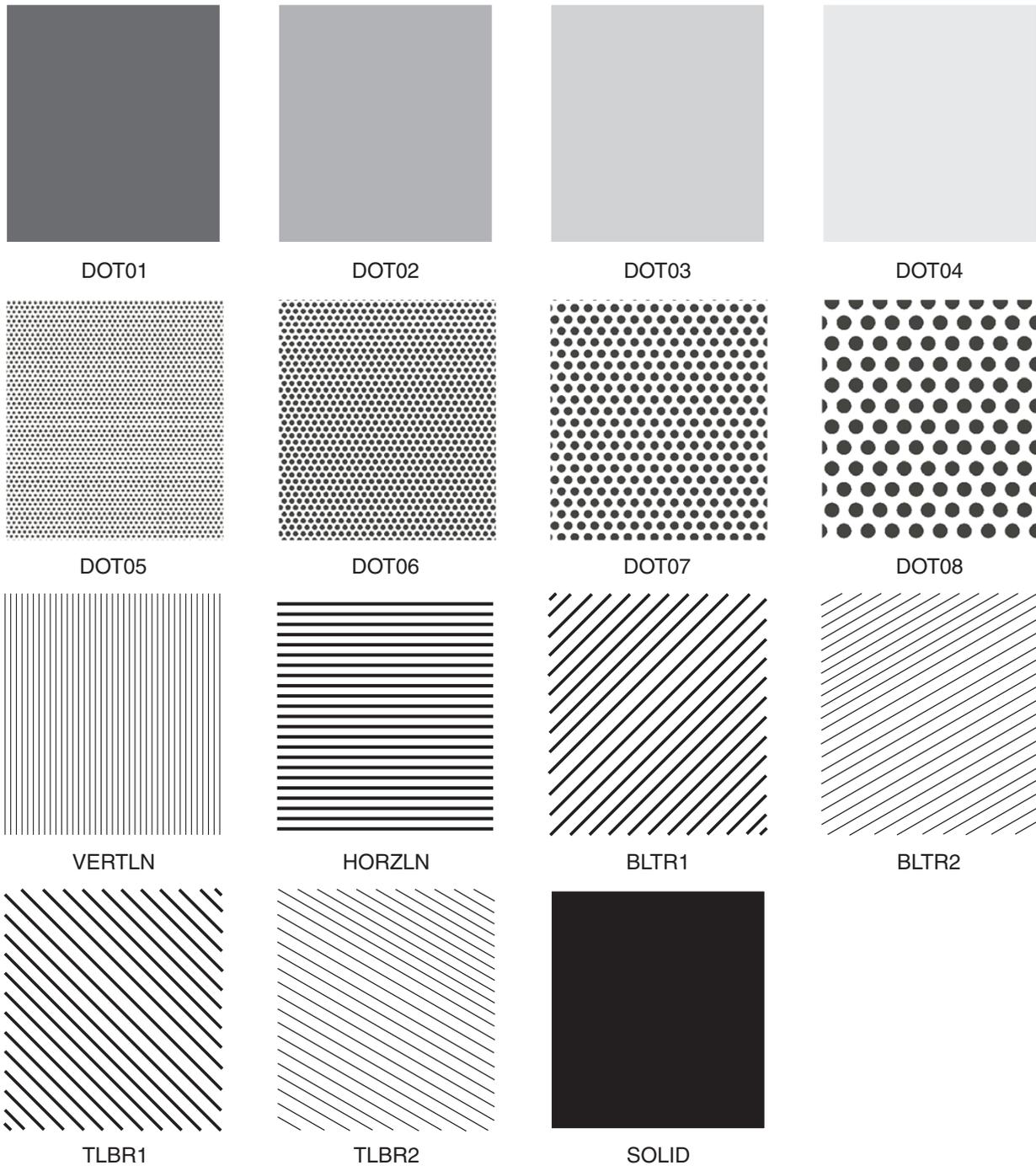


Figure 185. Fill Patterns

REPEAT Similar to existing **DRAWBOX** and **DRAWCIRCLE** processing. See “**DRAWBOX** Command” on page 206 and “**DRAWCIRCLE** Command” on page 217 for a detailed description of this parameter.

WITHTEXT Same as existing **DRAWBOX** and **DRAWCIRCLE** processing. See “**DRAWBOX** Command” on page 206 and “**DRAWCIRCLE** Command” on page 217 for a detailed description of this parameter.

Programming Samples

Draw a series of boxes:

```

SETUNITS 1 MM 1 MM ;

OVERLAY GOCBOX  SIZE 210 260  OFFSET 0 0 ;
CONTROL REPLACE ;
ORIENT 0 ;

DEFINE BLUE1 COLOR OCA BLUE;
DEFINE RED1  COLOR OCA RED;
DEFINE GREEN1 COLOR OCA GREEN;

POSITION 60 80 ;
drawgraphic box 20 10 3 dotted blue1
  repeat down 2 spaced 50 fill box 1 bltr2 red1 fill box 3 bltr1 green1;

POSITION 120 80 ;
drawgraphic box 20 10 rounded
  5 dshdbldot red1
  repeat down 2 spaced 50 fill box 2 blue1;

```

The following example overlay shows several types of objects:

```

SETUNITS 1 MM 1 MM ;

OVERLAY GOCALL  SIZE 210 260  OFFSET 0 0 ;
CONTROL REPLACE ;
ORIENT 0 ;

DEFINE BLUE1 COLOR OCA BLUE;

POSITION 60 80 ;
drawgraphic circle 30 blue1
  repeat down 2 spaced 20;

POSITION 80 100 ;
drawgraphic ellipse 100 185 80 100 ;

POSITION 20 150 ;
drawgraphic fillets to 10 15 to 100 15 ;

POSITION 100 10 ;
drawgraphic marker diamond ;

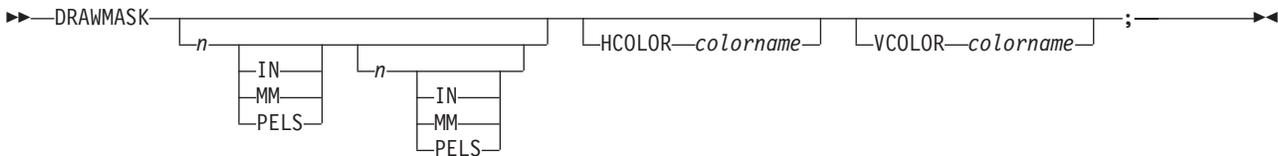
```

DRAWMASK Command

DRAWMASK Command

Use this command to draw a grid on the overlay as a design aid.

DRAWMASK Command



command word

DRAWMASK

first spacing

Using a number (*n*) and a unit of measurement, specify the distance from the left side of one vertical grid rule to the left side of the next vertical grid rule. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

second spacing

Using a number (*n*) and a unit of measurement, specify the distance from the top of one horizontal grid rule to the top of the next horizontal grid rule. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

HCOLOR and VCOLOR

Use **HCOLOR** and **VCOLOR** to color the horizontal and vertical rules that compose the mask. **HCOLOR** and **VCOLOR** should follow the mask spacing values.

If the **HCOLOR** color definition has not been previously defined, the horizontal rules will be done in the device default color. If the **VCOLOR** color definition has not been previously defined, the vertical rules will be done in the device default color.

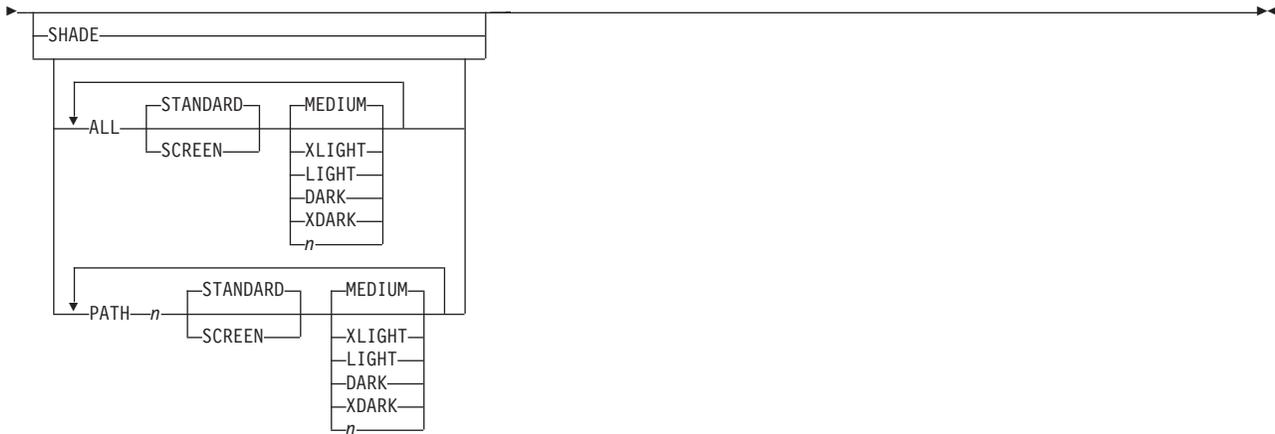
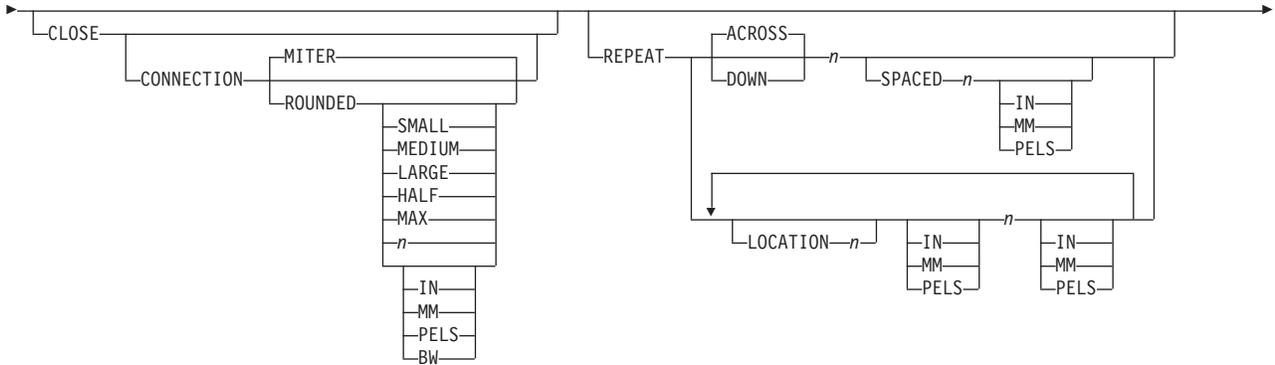
Notes:

1. You can draw a grid on your overlay by making no entry other than **DRAWMASK**. If you do this, the program defaults to the units of measurement established in the last **SETUNITS** command.
2. If you provide only one spacing value, the program uses that value to space both the horizontal and the vertical rules.
3. If you enter a number of units in the **DRAWMASK** command but do not specify a unit of measurement, the program multiplies the default value in **SETUNITS** by the number you supply and uses the resulting distance to space the rules of the grid.

end marker

Always end a command with an end marker (;).

DRAWPATH Command



command word

DRAWPATH

path thickness Specify the thickness of the path. The choices for path thickness are one of the following or a number (no decimals) indicating the thickness in pels. Choose from:

LIGHT 2 pels wide.
MEDIUM 4 pels wide (default).
BOLD 6 pels wide.
n Thickness in pels, do not add the word "pels".

Note: 0 thickness creates an invisible path, but text and shading can still be specified.

path type Specify the type of path you want. Choose from:

SOLID (Default)
DASHED
DOTTED

subcommand word

Enter **PATHEND** to specify the shape of the path start or end points. The path start point is the position from which the first path segment is drawn. The path end point is the position to which the last path segment is drawn.

path start shape

Choose from:

SQUARE Results in a rectangular end.
TRIMMED Results in the edges of the path end being trimmed to the horizontal and vertical. This means the path end is pointed.

ROUNDED Is rounded by a semicircle with its center at the specified path end point.

path end shape

Choose from:

SQUARE Results in a rectangular end.

TRIMMED Results in the edges of the path end being trimmed to the horizontal and vertical. This means the path end is pointed.

ROUNDED Is rounded by a semicircle with its center at the specified path end point.

The **PATHEND** option can be specified once or twice. If two **PATHEND** options are specified, the first applies to the path start, and the second applies to the path end. If only one **PATHEND** option is specified, it applies to both the start and end shapes. See “Trimmed Path Ends” on page 182 for more information on path ends.

Figure 186 illustrates the end shapes.

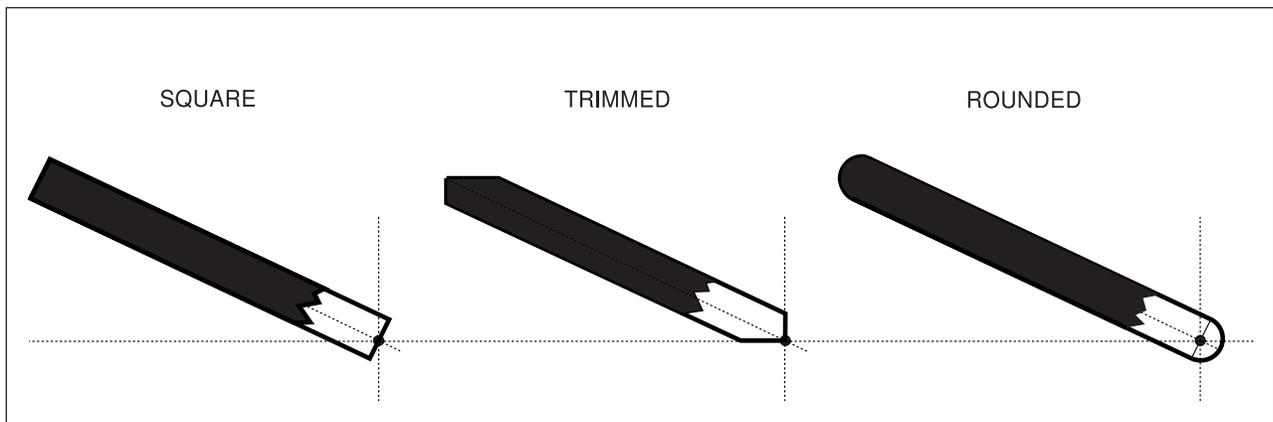


Figure 186. Path End Shapes

Note: The **PATHEND** subcommand is ignored if the path is closed (for example, it starts and ends at the same point).

subcommand word

Enter **CONNECTION** if you want to specify a default connection type for all connections made in this path.

connection type

Choose from:

MITER (Default)

ROUNDED

The connection types are illustrated in Figure 187 on page 246.

If the **ROUNDED** connection type is selected, the amount of rounding is controlled by the previous **SETUNITS** command. For an explanation of the length of rounded connections and connection length defaults, please refer to “Corner Length Values with **SETUNITS**” on page 141.

The connection type specified here is used as a default for all connections on this path. However, on individual points a connection type can be specified to override this default.

connection length

Specify the connection length if you want to set the default corner length for rounded connections in this path. Figure 130 on page 142 illustrates the corner length of a connection. Choose from:

DRAWPATH Command

SMALL
MEDIUM
LARGE
HALF
MAX

MAX gives a corner length that extends the full length of the shorter of the two segments being connected. **HALF** gives a corner length that extends $\frac{1}{2}$ that length.

Instead of a keyword, you can enter a number (n) and a unit of measurement:

n

IN	Inches
MM	Millimeters
PELS	Pels
BW	Multiples of the border weight

Note: If you specify a number of units with no unit of measurement, the program uses the unit of measurement value from the last primary default you specified with **SETUNITS**.

Notes:

1. Where **MITER** connections would be excessively long, OGL/370 automatically draws another kind of connection (called bevel). See Figure 187
2. **ROUNDED** connections may not actually touch the connection point. Observe how the **ROUNDED** connection where the angle is relatively small does not extend as far as the **MITER** (bevel) connection above it. This effect diminishes if the corner length is made smaller.

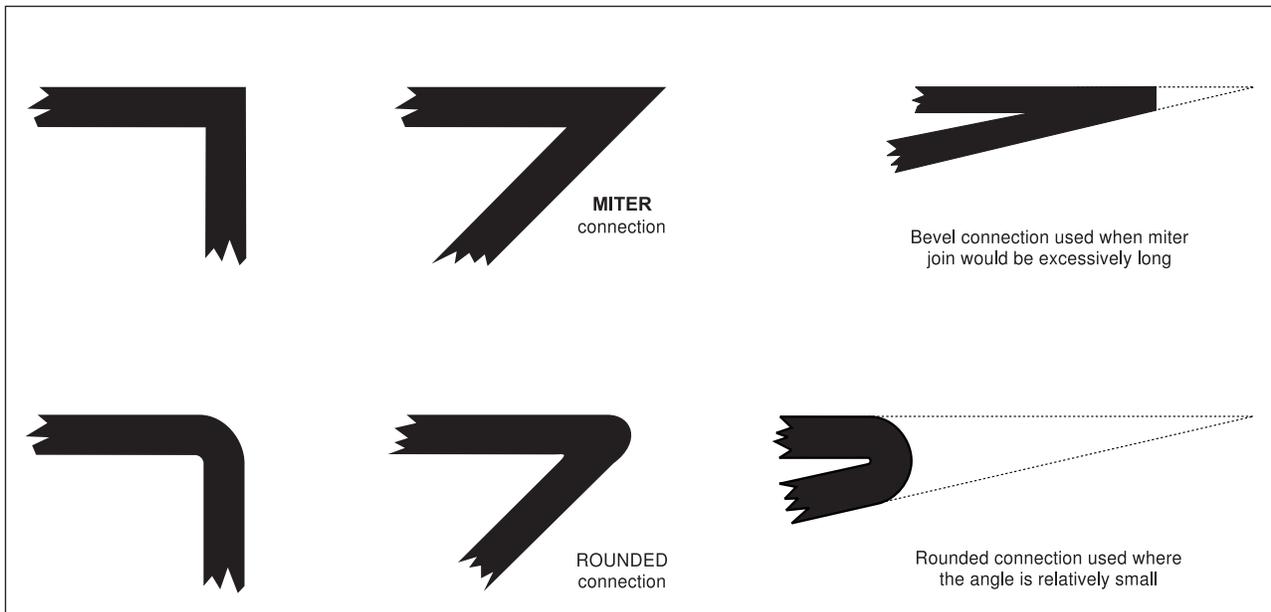


Figure 187. Path Connection Types

DRAWPATH Command - Specifying Path Points

A path is one or more straight lines connected between points. You control where the path goes by specifying the points on the path.

The path begins at the point specified by the last **POSITION** command preceding the **DRAWPATH** command. If no position command is specified before the **DRAWPATH** command, the path begins at the overlay origin.

The path is drawn, in turn, to each position specified with the **TO** subcommand.

subcommand word

Enter **TO** to specify the connection point for the next segment of the path. Repeat this subcommand for each point in the path.

origin option Specify which kind of positioning to use for the first coordinate:

Absolute positioning: The first coordinate is measured from the overlay (or **GROUP**) origin; enter **ABSOLUTE** (default).

Relative positioning: The first coordinate is measured from the previous point on the path; enter **LEFT**, **RIGHT**, **UP**, or **DOWN**.

first coordinate Using a number (*n*) and a unit of measurement, specify the first of two coordinates necessary to define your path point. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

origin option Specify which kind of positioning to use for the second coordinate:

Absolute positioning: The first coordinate is measured from the overlay (or **GROUP**) origin; enter **ABSOLUTE** (default).

Relative positioning: The first coordinate is measured from the previous point on the path; enter **LEFT**, **RIGHT**, **UP**, or **DOWN**.

second coordinate

Using a number (*n*) and a unit of measurement, specify the second coordinate necessary to define your path point. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for either of the coordinates, the program defaults to the current values in the **SETUNITS** command.

close option Enter **CLOSE** to connect the end of the path to the start of the path.

subcommand word

Specify **CONNECTION** to use the connection option for this point in the path.

connection type

Select the type of connection to be made at this point in the path. The connection types are:

MITER (Default)
ROUNDED

The selection here overrides the default path connection type for this point only.

DRAWPATH Command

connection length

Specify the connection length to set the corner length of a rounded connection at this point of the path. Choose from:

SMALL
MEDIUM
LARGE
HALF
MAX

Instead of a keyword, you can enter a number (*n*) and unit of measurement:

n

IN Inches
MM Millimeters
PELS Pels
BW Multiples of the border weight

Note: If you specify a number of units with no unit of measurement, the program uses the unit of measurement value from the last primary default you specified with **SETUNITS**. The corner length specified here overrides the default path corner length for this point only.

end marker

If you are drawing a single path without shading or text, enter an end marker (;) to complete the **DRAWPATH** command.

Notes:

1. You can mix an **ABSOLUTE** (default), with a **RIGHT**, **LEFT**, **DOWN**, or **UP** in the same **TO** subcommand. When **ABSOLUTE** is specified as the *first coordinate*, it is the horizontal coordinate. Specified as the *second coordinate*, **ABSOLUTE** is the vertical coordinate with respect to the overlay origin.
2. Two coordinates are required, even if one is a zero coordinate. Remember, specifying 0 without a relative keyword will result in **ABSOLUTE 0**.
3. If you specify only a number of units (without specifying a unit of measurement) for the coordinate entries, the program uses the unit of measurement values from the last **SETUNITS** command (not part of a **GROUP** definition).
4. When **DRAWPATH** occurs within a **GROUP** definition, all **ABSOLUTE** positioning is done with respect to the **GROUP** origin.

DRAWPATH Command-Spaced Repetition

There are two ways to repeat paths:

Spaced Repetition: Enter one set distance between paths

Location Repetition: Enter the location of each repeated path.

If the answer to both of the following questions is “yes”, use spaced repetition as described here. If the answer to either or both of the questions is “no”, see “**DRAWPATH** Command-Location Repetition” on page 250.

1. Are the path origins lined up either horizontally or vertically?
2. Is the spacing to be equal between the paths?

subcommand word

Specify **REPEAT** if you want to draw more than one path of the same size, shape, path thickness, and path type.

If spaced repetition is used, you may repeat any paths beside or below the original.

direction

Choose from:

ACROSS The paths are repeated beside the original path (default).

DOWN The paths are repeated below the original path.

repetitions

Enter the number (*n*) of additional paths.

spacing word

SPACED

spacing value

With a number (*n*) and a unit of measurement, specify the space you want between the starting points of the paths. Choose from:

n

IN Inches

MM Millimeters

PELS Pels

Note: There is no default spacing for paths.

DRAWPATH Command

DRAWPATH Command-Location Repetition

There are two ways to repeat paths:

Location Repetition: Enter the location of each repeated path.

Spaced Repetition: Enter one set distance between paths.

If the answer to either or both of the following questions is “no”, use location repetition as described here. If the answer to both of the following questions is “yes”, see “**DRAWPATH Command-Spaced Repetition**” on page 249.

1. Are the origins of the paths in either horizontal or vertical straight lines?
2. Is the spacing to be equal between the paths?

subcommand word

Specify **REPEAT** if you want to draw more than one path of the same size, path thickness, and path type.

location option Specify **LOCATION** to repeat the path anywhere within the overall dimensions of the overlay.

horizontal coordinate

With a number (*n*) and a unit of measurement, specify the distance from the left edge of the overlay (or **GROUP** if the command is within a **GROUP** definition) to the point where you want the repeated path to begin. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the horizontal coordinate, the program defaults to the current value in the **SETUNITS** command.

vertical coordinate

With a number (*n*) and a unit of measurement, specify the distance from the top of the overlay (or **GROUP** if the command is part of a **GROUP** definition) to the point where you want the repeated path to begin. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the vertical coordinates, the program defaults to the horizontal coordinate or the current spacing value in the **SETUNITS** command.

DRAWPATH Command-Shading Paths

You can shade the region enclosed by a path. However, path shading only takes place if the path is closed, that is, if the path starts and ends at the same point on the overlay.

shading option **SHADE**

path Choose from:

ALL The same shading in all of the paths created by the current **DRAWPATH** command (default).

PATH *n* Shade a particular path in the overlay.

Note: Using spaced repetition, paths are assigned numbers by their left to right or top to bottom order. If you used location repetition, path numbers are determined by the order in which they were defined.

shade pattern Choose from:

STANDARD The interior of the paths shaded with the pattern shown in Figure 211 on page 326 (default).

SCREEN The interior of the paths shaded with the pattern shown in Figure 212 on page 330.

shade type Specify how dark the shading should be for the interior of the paths.

To specify the amount of shading you want for the path interiors:

- Enter one of the standard choices illustrated in Appendix G, “Shade Patterns and Types” on page 325. **MEDIUM** is the default.
- If you prefer shading that is lighter or darker than one of the standard choices, enter a number from 0 to 100 that matches the percentage of shading you want. See Appendix G, “Shade Patterns and Types” on page 325 for examples of all percentages.

end marker Always end a command with an end marker (;).

DRAWPATH Command

DRAWPATH Command-Tips

The **DRAWPATH** command is a very flexible command that allows you to create almost any shape. To get the most out of **DRAWPATH**, you must be aware of the way it creates its connections. The two types of connections, **MITER** and **ROUNDED**, are shown in Figure 187 on page 246. The importance of understanding how these connections are used is best explained by the following figures.

Using MITER Connections

Example A in Figure 188 shows four points used in the following **DRAWPATH** command:

```
DRAWPATH 40 SOLID CONNECTION MITER
TO RIGHT 2 IN DOWN 0 IN
TO RIGHT 0 IN DOWN 1 IN
TO LEFT 2 IN UP 1 IN
TO LEFT 0 IN DOWN 1 IN
TO RIGHT 2 IN DOWN 0 IN;
```

Example B shows the results of the command. Example C shows how the **MITER** connection causes protrusions on the top-left and bottom-right corners of the box. This effect may happen whenever a path uses the same point twice, but with two different angles of connection. Each time a point is used, a **MITER** connection is formed by the lines connecting at that point.

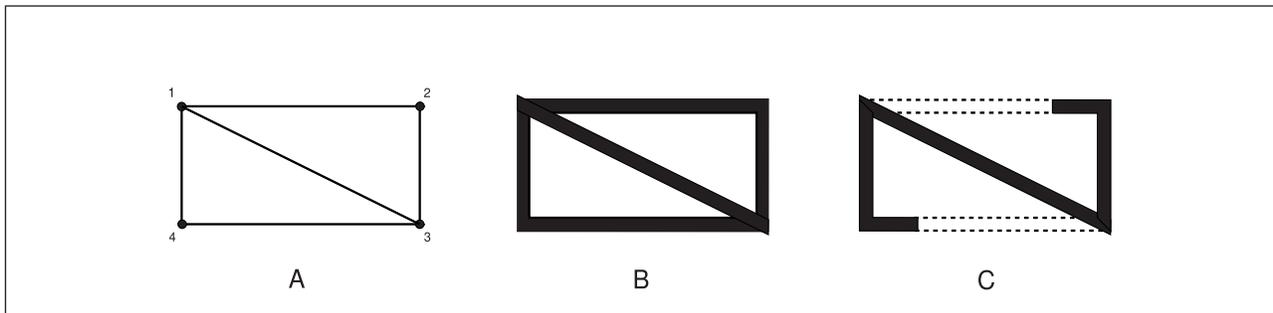


Figure 188. Examples of **MITER** Connections

Using ROUNDED Connections

Figure 189 on page 253 illustrates the differences between **ROUNDED** and **MITER** connections.

In example D of Figure 189 on page 253 the connections alternate between **ROUNDED MAX** and **MITER**.

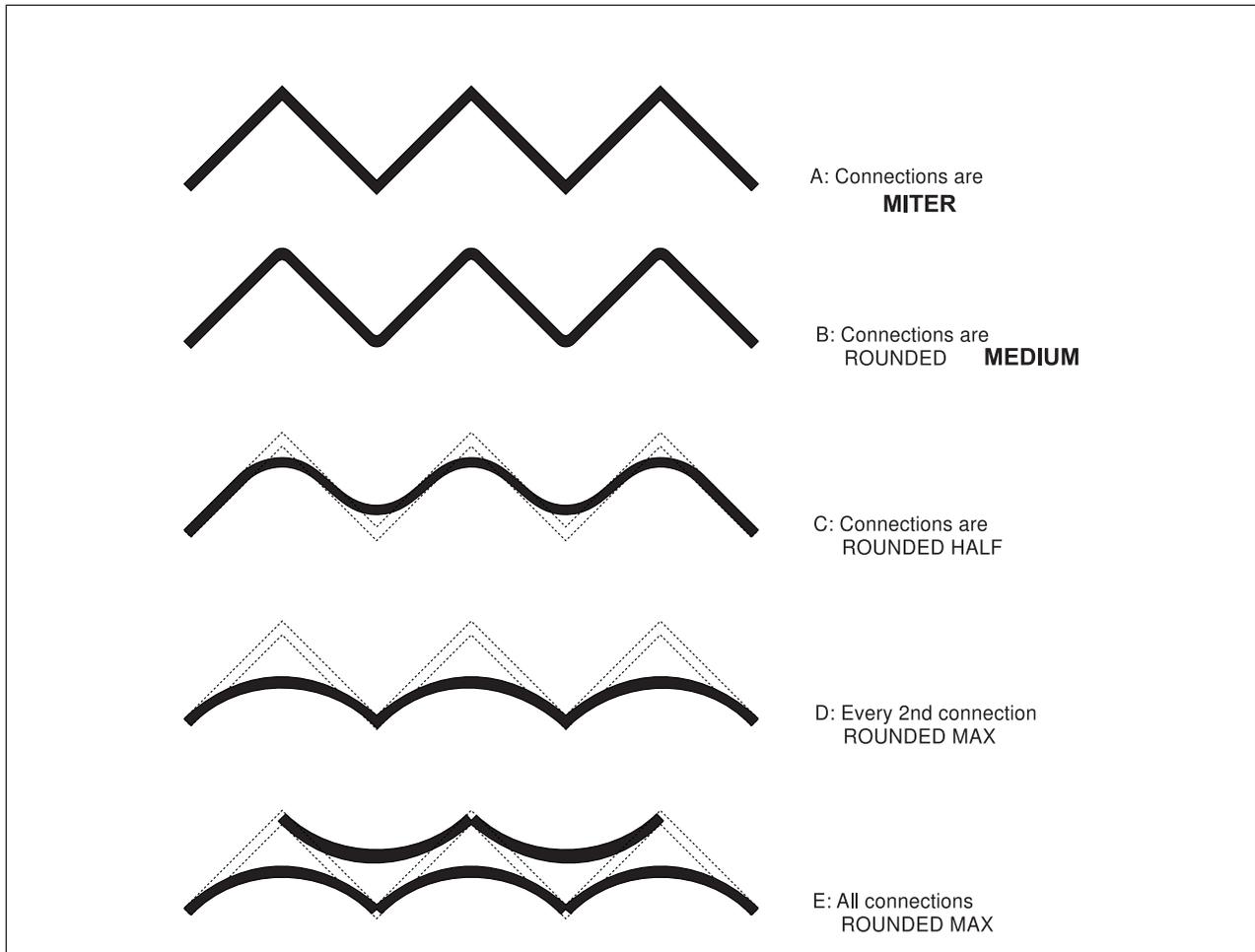


Figure 189. Examples of **ROUNDED** Connections

The **DRAWPATH** command used to create example D is:

```

DRAWPATH 20 SOLID CONNECTION ROUNDED MAX
TO RIGHT 150 PELS UP 150 PELS
TO RIGHT 150 PELS DOWN 150 PELS CONNECTION MITER
TO RIGHT 150 PELS UP 150 PELS
TO RIGHT 150 PELS DOWN: 150 PELS CONNECTION MITER
TO RIGHT 150 PELS UP 150 PELS
TO RIGHT 150 PELS DOWN 150 PELS;

```

⋮

Shading Closed Paths

Figure 190 on page 254 shows three similar paths; the only difference between them is the rounding used in the connections. Path 1 has none of its connections rounded. The other two examples show how two adjacent rounded connections can cause a path to be disjointed.

In Path 2, points A and B show the ends of the rounded connections. Due to the size of the rounding, the ends of the arcs do not meet and the path becomes disjointed. Note that the actual points specified form a closed path, but because it is disjointed, it cannot be shaded.

DRAWPATH Command

Path 3 shows how tricky disjointed paths can be. The path appears closed because the line thickness hides the fact that the two arcs do not join. The ends of the arcs are very close together, but A is still to the left of B, just as in Path 2. To close the path, the rounding must be reduced to **ROUNDED HALF** or smaller, so that A is to the right of B; only then may the path be shaded.

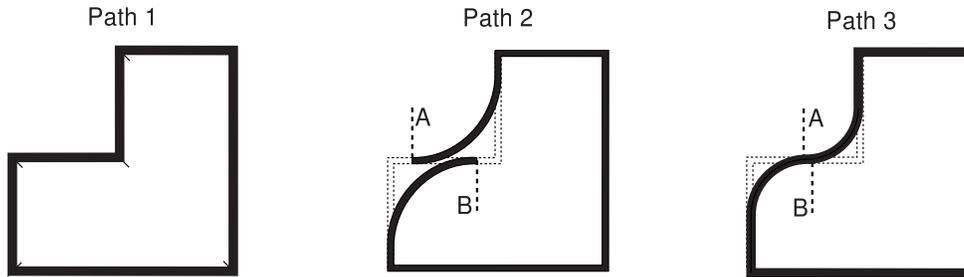


Figure 190. Examples of Closed and Disjointed Paths

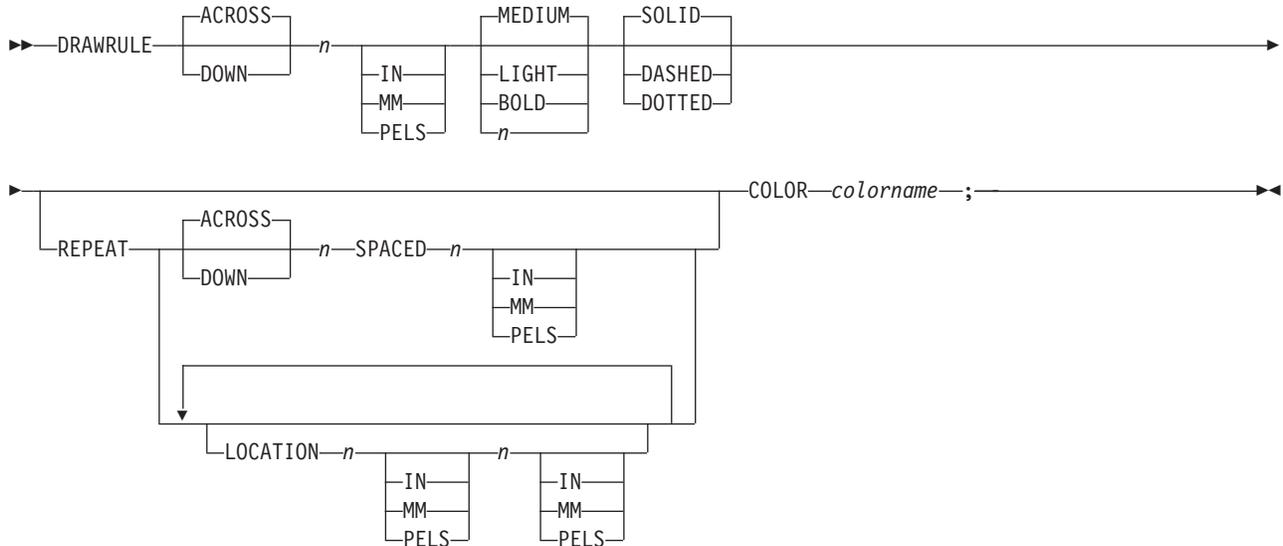
The **DRAWPATH** command used to create Path 2 is:

```
DRAWPATH 10 SOLID CONNECTION MITER
  TO RIGHT 240 PELS UP 0 PELS CONNECTION ROUNDED 200
  TO RIGHT 0 PELS UP 240 PELS
  TO RIGHT 240 PELS UP 0 PELS
  TO RIGHT 0 PELS DOWN 480 PELS
  TO LEFT 480 PELS DOWN 0 PELS
  TO LEFT 0 PELS UP 240 PELS CONNECTION ROUNDED 200;
```

DRAWRULE Command

Use this command to draw a horizontal or vertical rule or line on your overlay.

DRAWRULE Command



command word

DRAWRULE

rule direction

Choose from:

ACROSS (Default)
DOWN

The positioning of rules is controlled by the **SETUNITS** command. See “Top-Left and Center Positioning with **SETUNITS**” on page 144 for more information.

rule length

Using a number (n) and a unit of measurement, specify the length of the rule you want to draw. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

Note: If you do not specify a unit of measurement, the program defaults to the current value in the **SETUNITS** command.

rule thickness

Specify the thickness of the rule. You can specify a number (n) and a unit of measurement (in pels), or a ready-made thickness. Choose from:

LIGHT 2 pels wide.
MEDIUM 4 pels wide.
BOLD 6 pels wide.
 n Thickness in pels, do not add the word “pels”.

rule type

Specify the type of rule you want to draw. Choose from:

SOLID (Default)
DASHED
DOTTED

DRAWRULE Command

COLOR Use **COLOR** to specify the color for the rule. **COLOR** must follow the **REPEAT** subcommand.

If the color definition has not been previously defined, the rule will be done in the device default color.

DRAWRULE Command-Spaced Repetition

There are two ways to repeat rules:

Spaced Repetition: Enter one set distance between rules.

Location Repetition: Enter the location of each repeated rule.

If the answer to both of the following questions is “yes”, use spaced repetition as described here. If the answer to either or both of the following questions is “no”, see “**DRAWRULE** Command-Location Repetition” on page 257.

1. Are the rule origins lined up either horizontally or vertically?
2. Is the spacing to be equal between the rules?

subcommand word

Specify **REPEAT** if you want to draw additional rules of the same length, thickness, and type as the one you just specified.

If spaced repetition is used, you may repeat any rules beside or below the original.

direction

Choose from:

ACROSS The repeated rule or rules appear beside the original rule (default).
DOWN The repeated rule or rules appear below the original rule.

repetitions

Enter the number (*n*) of additional rules.

spacing word

SPACED

spacing value

With a number (*n*) and a unit of measurement, specify how much space you want between rules. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

Notes:

1. Use Table 2 as a guide to measuring the space between rules.
2. If you do not specify a unit of measurement, the program defaults to the current value in the **SETUNITS** command.

Table 2. Spacing Value

Original Line	Repeated Line	Measure
ACROSS	ACROSS	From the end of one rule to the beginning of the next rule.
ACROSS	DOWN	From the top of one rule to the top of the next rule.
DOWN	ACROSS	From the left edge of one rule to the left edge of the next rule.
DOWN	DOWN	From the bottom of one rule to the top of the next rule.

end marker Use an end marker (;) when you have finished making entries for Spaced Repetition rules.

DRAWRULE Command-Location Repetition

There are two ways to repeat rules:

Location Repetition: Enter the location of each repeated rule.

Spaced Repetition: Enter one set distance between rules.

If the answer to either or both of the following questions is “no”, use location repetition as described here. If the answer to both of the following questions is “yes”, see “**DRAWRULE Command-Spaced Repetition**” on page 256.

Notes:

1. Are the starting positions for all of the rules to be horizontal or vertical straight lines?
2. Is the spacing to be equal between the rules?

subcommand word

Specify **REPEAT** if you want to draw additional rules of the same length, thickness, and type as the one you just specified.

location option Specify **LOCATION** to repeat the rule anywhere within the overall dimensions of the overlay.

horizontal coordinate

Using a number (*n*) and a unit of measurement, specify the distance from the left edge of the overlay (or **GROUP** origin if the command is within a **GROUP** definition) to the point where you want the repeated rule to begin. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the horizontal coordinate, the program defaults to the current value in the **SETUNITS** command.

vertical coordinate

Using a number (*n*) and a unit of measurement, specify the distance from the top of the overlay (or **GROUP** origin if the command is part of a **GROUP** definition) to the point where you want the repeated rule to begin. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a unit of measurement for the vertical coordinate, the program defaults to the horizontal coordinate or the current value in the **SETUNITS** command.

end marker Always end a command with an end marker (;).

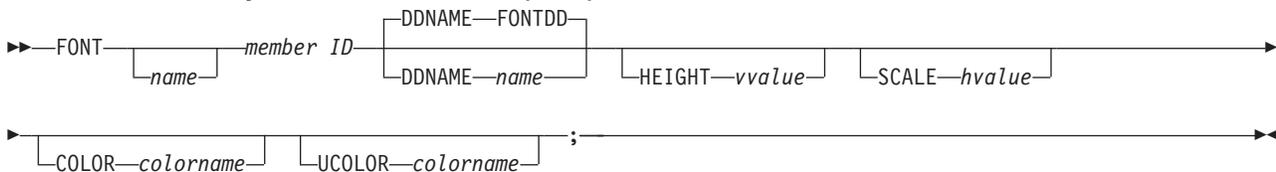
FONT Command (MVS)

FONT Command (MVS)

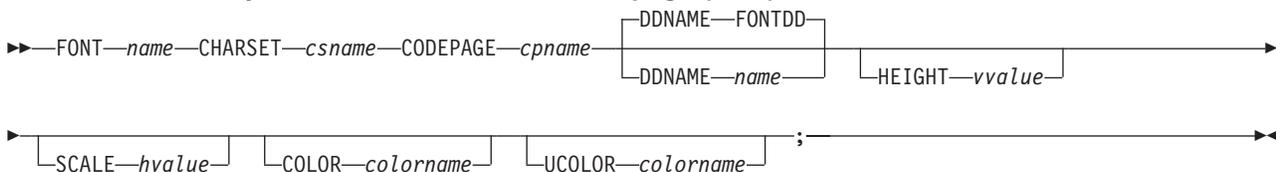
In the MVS environment, use this command to specify the font, or fonts, used to print the text in the overlay.

Refer to *ABOUT TYPE: IBM's Technical Reference for 240-Pel Digitized Type* and *ABOUT TYPE: IBM's Technical Reference for Core Interchange Fonts* for information about fonts usable with OGL/370. For information about your company's special fonts, see your system programmer.

FONT Command Syntax with MemberID (MVS)



FONT Command Syntax with Character Set/Codepage (MVS)



command word

FONT

font name This entry is optional. Use this name in the **DRAWBOX**, **DRAWCIRCLE** (both with **WITHTEXT** entry) or **SETTEXT** command to specify the font.

The font name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

Note: If you do not enter a font name, the *member ID* entry of this command is used as the font name.

member ID Specify the *member ID* of the coded font that you want to use.

Notes:

1. The *member ID* is derived from the member name under which the font is stored in the font library. To derive the *member ID*, remove the two-character prefix from the member name. For example, the font for "TropiCal" in Figure 12 on page 14, is stored in the font library under the member name "X1BITR". To get this font in your overlay, you would specify "BITR" as your *member ID* entry.
2. Ask your system programmer for more information on font usage at your installation.

CHARSET/CODEPAGE

Character set and codepage names including prefixes.

DD statement word and statement name

Specify the *DD statement name*, if you want something other than **FONTDD**. The *DD statement name* is specified in the JCL. If the name is not **FONTDD**, use the first choice below:

DDNAME *name*

Enter **DDNAME** *name*, to specify a DD statement name other than **FONTDD**, where *name* is the name you have chosen.

DDNAME **FONTDD**

(Default)

HEIGHT

vvalue is the height and point size within one decimal place. This value is only valid for outline fonts.

SCALE

hvalue is the horizontal scaling factor to be applied to an outline font. For example, **SCALE** 10 means to scale a font to 10% of its current width.

COLOR/UCOLOR

Use **COLOR** to specify the color definition for text on **SETTEXT** or **WITHTEXT** subcommand. Use **UCOLOR** to specify the color definition for the text underscore.

If **COLOR** or **UCOLOR** are specified without a color definition name, the text or underscore will be rendered in the device default color.

COLOR and **UCOLOR** must follow **HEIGHT** and **SCALE**.

end marker

Always end a command with an end marker (;).

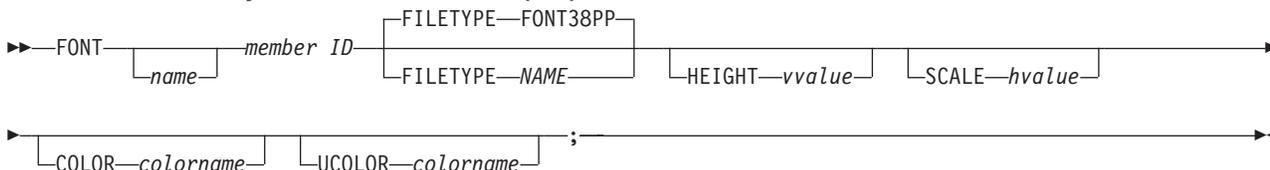
FONT Command (VM)

FONT Command (VM)

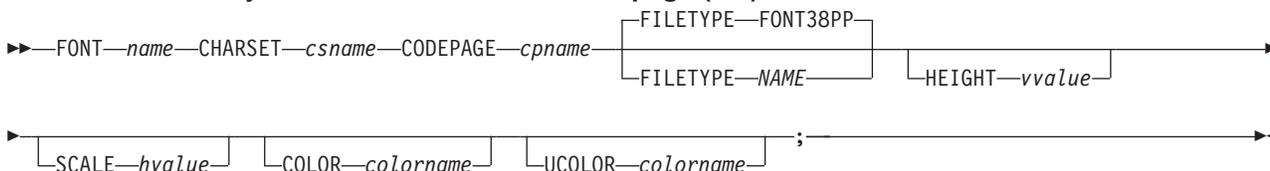
In the VM environment, use this command to specify the font, or fonts, used to print the text in the overlay.

Refer to *ABOUT TYPE: IBM's Technical Reference for 240-Pel Digitized Type* and *ABOUT TYPE: IBM's Technical Reference for Core Interchange Fonts* for information about fonts usable with OGL/370. For information about your company's special fonts and how to access them, see your system programmer.

FONT Command Syntax with MemberID (VM)



FONT Command Syntax with Character Set/Codepage (VM)



command word

FONT

font name This entry is optional. Use this name in the **DRAWBOX**, **DRAWCIRCLE** (both with **WITHTEXT** entry) or **SETTEXT** command to specify the font.

The font name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

Note: If you do not make a font name entry, the *member ID* entry of this command is used as the font name.

member ID Specify the *member ID* of the coded font that you want to use.

Note: The *member ID* is derived from the file name under which the font is stored. To derive the *member ID*, remove the two-character prefix from the filename of the coded font. For example, the font for "TropiCal" in Figure 12 on page 14, is stored in a file with a filename of "X1BITR". To get this font in your overlay, you would specify "BITR" as your *member ID* entry.

CHARSET/CODEPAGE

Character set and codepage names including prefixes.

filetype indicator and filetype name

Specify the **FILETYPE** name if your filetype is something other than **FONT38PP**. The filetype name is specified in the JCL. If the name is not **FONT38PP**, use the first choice below:

FILETYPE *name*

Enter **FILETYPE** *name*, to specify a filetype name other than **FONT38PP**.

FILETYPE FONT38PP

(Default)

HEIGHT

vvalue is the height and point size within one decimal place. This value is only valid for outline fonts.

SCALE

hvalue is the horizontal scaling factor to be applied to an outline font. For example, SCALE 10 means to scale a font to 10% of its current width.

COLOR/UCOLOR

Use **COLOR** to specify the color definition for text on **SETTEXT** or **WITHTEXT** subcommand. Use **UCOLOR** to specify the color definition for the text underscore.

If **COLOR** or **UCOLOR** are specified without a color definition name, the text or underscore will be rendered in the device default color.

COLOR and **UCOLOR** must follow **HEIGHT** and **SCALE**.

end marker

Always end a command with an end marker (;).

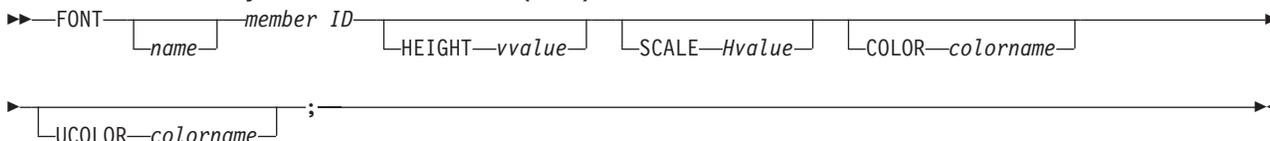
FONT Command (VSE)

FONT Command (VSE)

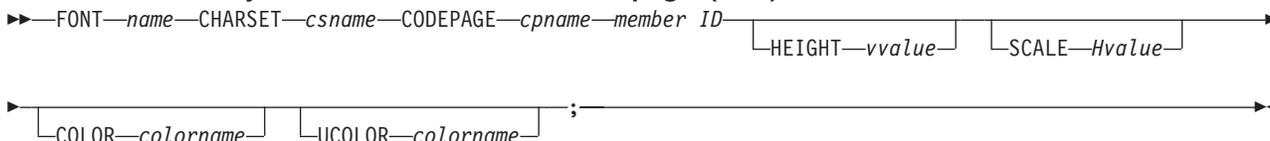
In the VSE environment, use this command to specify the font, or fonts, used to print the text in the overlay.

Refer to *ABOUT TYPE: IBM's Technical Reference for 240-Pel Digitized Type* and *ABOUT TYPE: IBM's Technical Reference for Core Interchange Fonts* for information about fonts usable with OGL/370. For information about your company's special fonts and how to access them, see your system programmer.

FONT Command Syntax with MemberID (VSE)



FONT Command Syntax with Character Set/Codepage (VSE)



command word

FONT

font name This entry is optional. Use this name in the **DRAWBOX**, **DRAWCIRCLE** (both with **WITHTEXT** entry) or **SETTEXT** command to specify the font.

The font name must meet these requirements:

1. The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

2. The name cannot be longer than six characters.

Note: If you do not specify a font name, the *member ID* of this command is used as the font name.

member ID Specify the *member ID* of the coded font that you want to use.

Notes:

1. The *member ID* is derived from the member name under which the font is stored in the font library. To derive the *member ID*, remove the two-character prefix from the member name. For example, the font for "TropiCal" in Figure 12 on page 14, is stored in the font library under the member name "X1BITR". To get this font in your overlay, you would specify "BITR" as your *member ID* entry.
2. Ask your system programmer for more information on font usage at your installation.

CHARSET/CODEPAGE

Character set and codepage names including prefixes.

FONT Command (VSE)

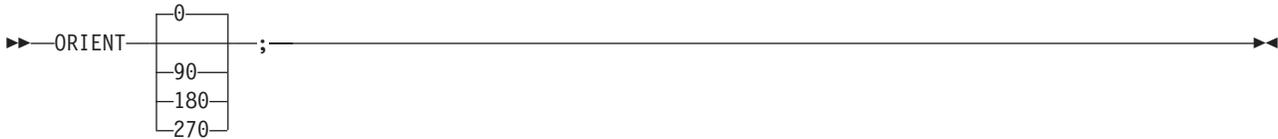
- HEIGHT** *vvalue* is the height and point size within one decimal place. This value is only valid for outline fonts.
- SCALE** *hvalue* is the horizontal scaling factor to be applied to an outline font. For example, SCALE 10 means to scale a font to 10% of its current width.
- COLOR/UCOLOR**
Use **COLOR** to specify the color definition for text on **SETTEXT** or **WITHTEXT** subcommand. Use **UCOLOR** to specify the color definition for the text underscore.
If **COLOR** or **UCOLOR** are specified without a color definition name, the text or underscore will be rendered in the device default color.
COLOR and **UCOLOR** must follow **HEIGHT** and **SCALE**.
- end marker* Always end a command with an end marker (;).

ORIENT Command

ORIENT Command

Use this command to specify the printing orientation of the overlay. You design the overlay in the direction it is to be read, but with this command you can instruct the program to rotate the overlay before printing.

ORIENT Command



command word

ORIENT

orientation

Specify the number of degrees you want the overlay rotated for printing (rotation is always clockwise). Choose from:

0 (Default)

90

180

270

Orientation Considerations:

1. Segments — a segment cannot be rotated and can be printed only in the orientation in which it was designed. See any of the following manuals for more information about creating segments for use with OGL/370:
 - *Print Management Facility User's Guide and Reference*
 - *Page Printer Formatting Aid User's Guide and Reference*
 - *Graphical Data Display Manager: General Information*
2. Text orientation — some printers have restrictions on what print directions are allowed. See the publications for your printer to determine what orientations are available for your printer. Be sure when you rotate an overlay that the text you specified remains in an orientation and format that can be printed. See “Text Direction Restrictions” on page 335 for information about non-printable combinations of format and orientation.
3. The effect of the **ORIENT** command on overlay origin is illustrated in:
 - Figure 191 on page 265
 - Figure 192 on page 265
 - Figure 193 on page 266
 - Figure 194 on page 266

end marker

Always end a command with an end marker (;).

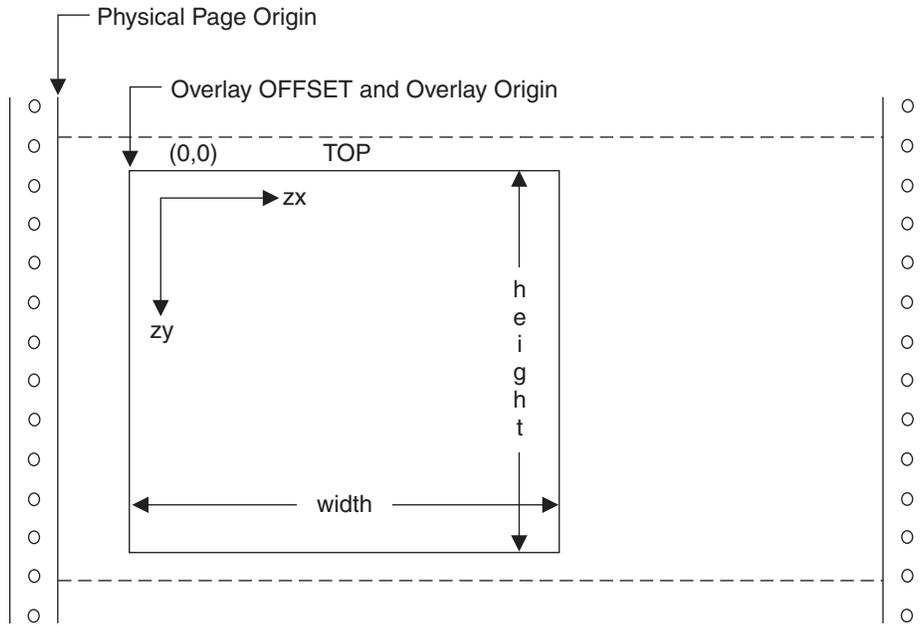


Figure 191. Overlay Offset and Overlay Origin for **ORIENT 0** Overlay

Note: The Overlay **OFFSET** and Overlay **ORIGIN** are at the same position. “zx” and “zy” are the horizontal and vertical coordinate directions respectively.

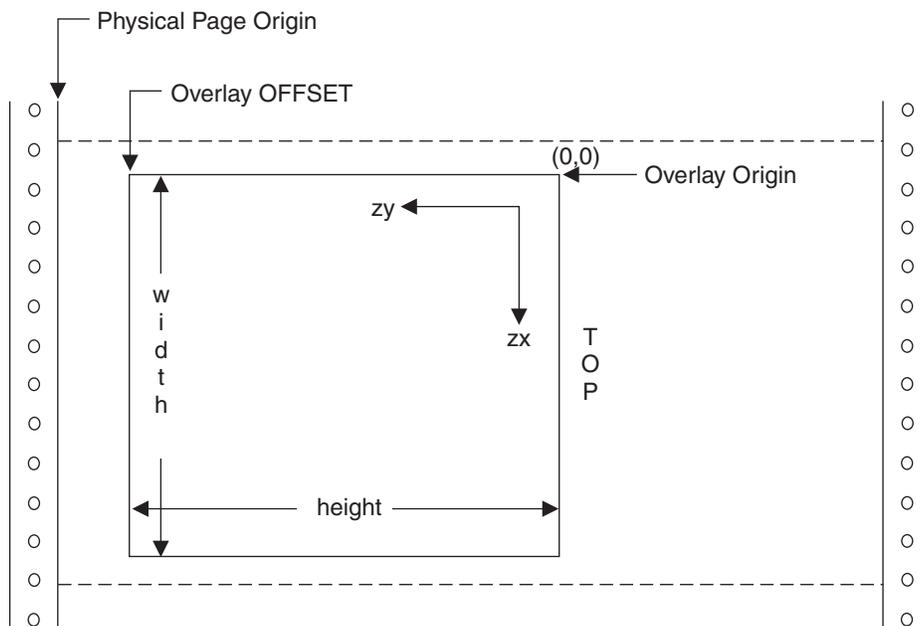


Figure 192. Overlay Offset and Overlay Origin for **ORIENT 90** Overlay

Note: The Overlay **OFFSET** and Overlay **ORIGIN** are at different positions. “zx” and “zy” are the horizontal and vertical coordinate directions respectively.

ORIENT Command

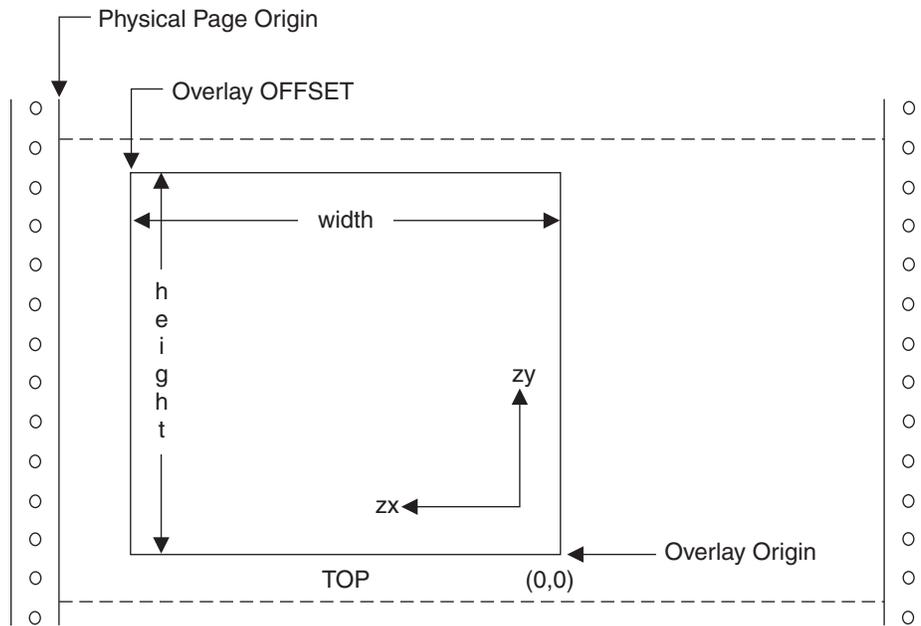


Figure 193. Overlay Offset and Overlay Origin for **ORIENT 180** Overlay

Note: The Overlay **OFFSET** and Overlay **ORIGIN** are at different positions. “zx” and “zy” are the horizontal and vertical coordinate directions respectively.

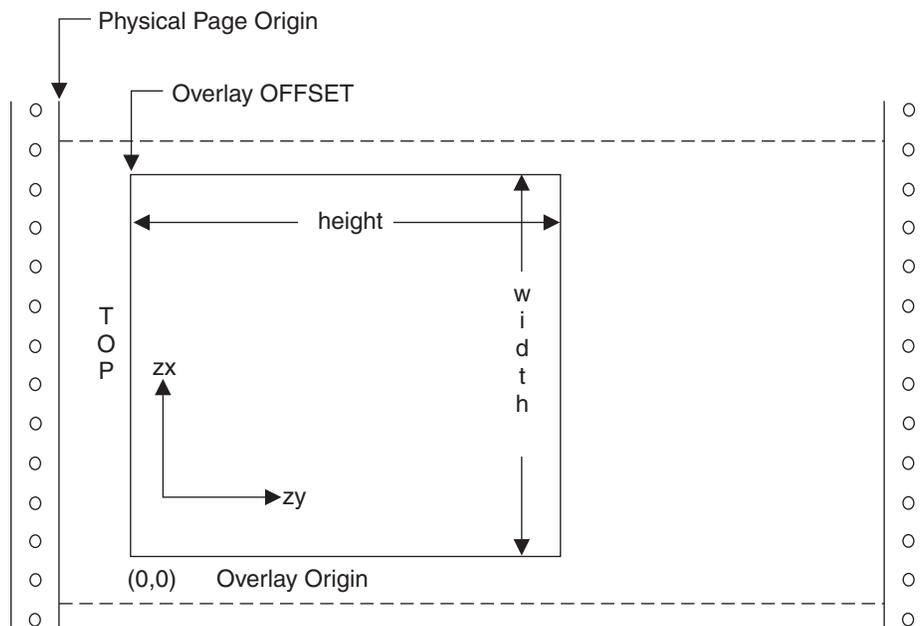


Figure 194. Overlay Offset and Overlay Origin for **ORIENT 270** Overlay

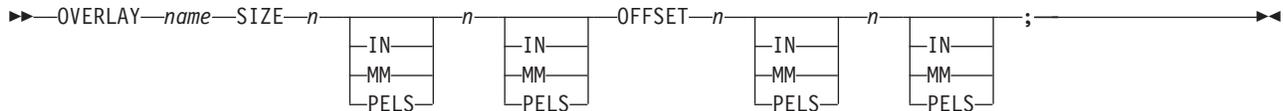
Note: The Overlay **OFFSET** and Overlay **ORIGIN** are at different positions. “zx” and “zy” are the horizontal and vertical coordinate directions respectively.

OVERLAY Command

Use this command to begin specifying the overall dimensions of your overlay and where it starts on the paper.

Note: If you are going to have a box surrounding the entire overlay (using the **DRAWBOX** command), the width and height dimensions you supply here must be larger than the dimension of the box by at least the thickness of one border.

OVERLAY Command



command word

OVERLAY

overlay name The font name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

subcommand word

Enter **SIZE** to set the dimensions of the overlay.

overlay width With a number (*n*) and a unit of measurement, specify the width of your overlay. Choose from:

<i>n</i>	IN	Inches
	MM	Millimeters
	PELS	Pels

overlay height With a number (*n*) and a unit of measurement, specify the height of your overlay. Choose from:

<i>n</i>	IN	Inches
	MM	Millimeters
	PELS	Pels

Note: If you do not specify a unit of measurement, the program defaults to the current value in the **SETUNITS** command.

subcommand word

Enter **OFFSET** to specify the coordinates of the overlay corner, nearest to the paper origin.

Notes:

1. Remember that the **OFFSET** point is relative to the physical position of the overlay on the page. OGL/370 automatically determines the overlay origin for you. Once this is done, all positions are with respect to the overlay origin. Thus, altering the **OFFSET** point makes no difference to the design of the overlay.
2. The overlay must fit onto the printable area of the page.

OVERLAY Command

horizontal coordinate

Using a number (*n*) and a unit of measurement, specify the horizontal distance from the paper origin to the nearest overlay corner. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a horizontal coordinate, the program defaults to the current value in the **SETUNITS** command.

vertical coordinate

Using a number (*n*) and a unit of measurement, specify the vertical distance from the top edge of the paper to the nearest overlay corner. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels

Note: If you do not specify a vertical coordinate, the program defaults to the current value in the **SETUNITS** command.

end marker Always end a command with an end marker (;).

See the following references for more information on the **OFFSET** subcommand:

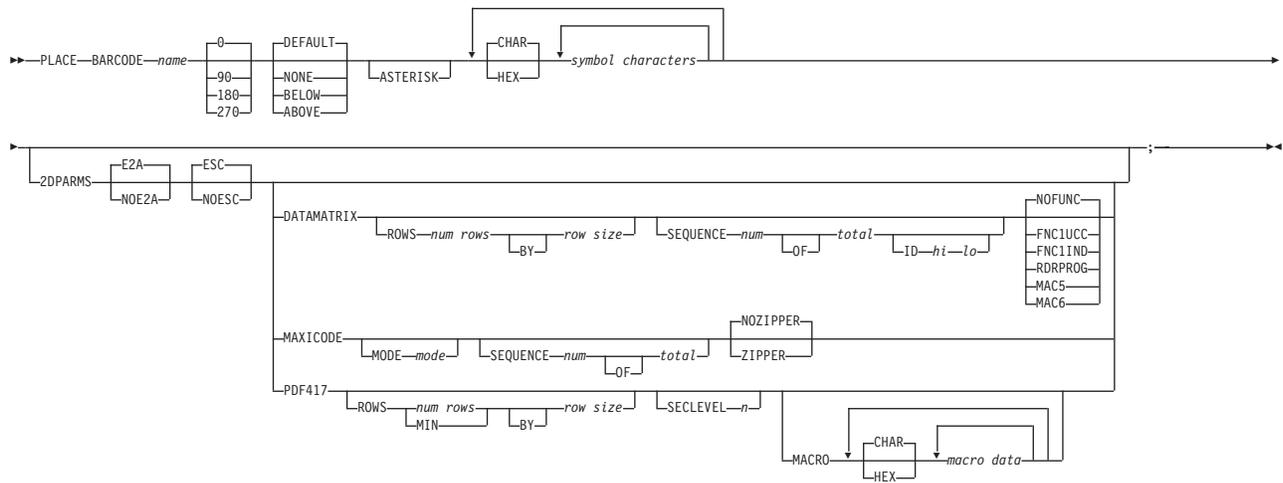
- “Beginning the Overlay (**OVERLAY**)” on page 27
- Figure 19 on page 30
- Figure 20 on page 31

PLACE BARCODE Command

Specify this command to place barcode objects on the overlay.

Note: This command generates BCOCA drawing orders that may require microcode support in your printer or PSF support in order to to print.

PLACE BARCODE Command



command word

PLACE

subcommand word

BARCODE

name The *name* used in the **DEFINE BARCODE** definition.

orientation Specify the orientation of the barcode. Choose from:
0 (Default)

90

180

270

HRI position Defines where the HRI characters are printed in relation to the barcode.

ASTERISK Causes the start and stop characters in **CODE39** barcodes to be shown as asterisks in the HRI. It is ignored in other symbologies. The default is no asterisks.

text type Specify the text type used in the string of barcode characters:

CHAR The characters you type are the exact characters that define the symbol (default).

HEX The text string to follow is in hexadecimal form.

symbol characters

2DPARMS This is an optional parameter for specifying two-dimensional barcodes (Data Matrix, MaxiCode, and PDF417 two-dimensional barcodes).

Translation

E2A

EBCDIC to ASCII translation. The printer converts each byte of the data from EBCDIC codepage 500 to the

PLACE BARCODE Command

appropriate ASCII codepage. See *Bar Code Object Content Architecture Reference*, S544-3766-04 or later for more details. This is the default if neither is coded.

NOE2A No translation.

Escape Sequence

ESC Escape Sequence Handling. This is the default if neither is coded. Each backslash character within the bar code data is treated as an escape character. Note that in this case no code page switching can occur within the data if **E2A** is selected.

NOESC Ignore Escape Sequences. Each backslash character within the bar code data is treated as a normal character. No code page switching can occur within the data.

DATAMATRIX Selects the two-dimensional Data Matrix bar code type.

Rows Defines the bar code symbol dimensions with defaults if the dimensions are not specified.

ROWS The number of rows and the number of modules per row are checked for the validity of the combination.

num rows The number of rows in the symbol. The default is **0**.

BY Optional parameter for readability.

size The desired number of modules in each row of the bar code symbol. The default is **0**.

Sequence The keywords for the placement of the bar code symbol in a sequence of bar code symbols. The default for the symbol is to stand alone.

SEQUENCE Indicates that the sequence of bar code parameters will follow.

num The sequence number of the bar code symbol. The default is **0**.

OF Optional parameter for readability.

total The total number of bar code symbols in the sequence in a range from 2—16. The default is **0**.

File ID The identifier used to match the set of structured append symbols of which this barcode symbology is a component. The identifier is specified by high and low order bytes that must be in the range of 1 to 254.

ID Indicates that the *fileid* parameters follow.

hi The high byte. The default is **1**.

lo The low byte. The default is **1**.

Data Matrix Special Functions

These are special functions which can only be used with a Data Matrix symbol. If not coded, the default is **NOFUNC** (user defined symbol).

- NOFUNC** No special function is used.
- FNC1UCC** UCC/EAN1 alternate data type identifier. A FNC1 is added in the first data position (or fifth position of a structured append symbol) to indicate that this bar code symbol conforms to the USS/EAC application identifier standard format.
- FNC1IND** Industry FNC1 alternate data type identifier. An FNC1 is added in the second data position (or sixth data position of a structured append symbol) to indicate that this bar code symbol conforms to a particular industry standard format.
- RDRPROG** Use this when the symbol contains a message used to program the barcode reader. In this case the barcode symbol cannot be a part of a structured append sequence.
- MAC5** This provides instructions to the bar code reader to insert an industry specific header and trailer around the symbol data. The bar code symbol contains a 05 Macro codeword. The barcode symbol cannot be a part of a structured append sequence.
- MAC6** Same as **MAC5** except the bar code symbol contains a 06 Macro codeword. The barcode symbol cannot be a part of a structured append sequence.

MAXICODE Selects the two-dimensional MaxiCode bar code type.

- MODE** *num* Symbol mode (used for MaxiCode two-dimensional barcode only). If not coded, the default is Standard Symbol Mode 4.
- 2** Structured Carrier Message — numeric postal code
 - 3** Structured Carrier Message — alphanumeric postal code
 - 4** Standard symbol (default)
 - 6** The bar code data is used to program the bar code reader system.

SEQUENCE Indicates that the sequence of bar code parameters will follow.

num The sequence number of the bar code symbol. The default is **0**.

OF Optional parameter for readability.

PLACE BARCODE Command

		<i>total</i>	The total number of bar code symbols in the sequence in a range from 2—8. The default is 0 .
		Zipper Pattern	Print a zipper pattern and contrast block (use for MaxiCode two-dimensional barcode only).
		<u>NOZIPPER</u>	Does not print a zipper pattern (default).
		ZIPPER	Prints a zipper pattern.
	PDF417		Selects the two-dimensional PDF417 bar code type.
		Rows	Defines the bar code symbol dimensions with defaults if the dimensions are not specified. The defaults allow the printer to decide based on the amount of symbol data.
		ROWS	The number of rows and the number of modules per row are individually checked for validity. The multiple of the two parameters must not exceed 928.
		<i>num rows</i>	The number of rows in the symbol which must be 255 or in the range of 3 to 90. The default is MIN or 255.
		BY	Optional for what for readability.
		<i>size</i>	The desired number of modules in each row of the bar code symbol. The default is 10 .
		SECLEV <i>n</i>	This parameter specifies the desired security level for the symbol as a value from 0 to 8 . Each higher security level causes more error correction codewords to be added to the symbol (used for PDF417 two-dimensional barcode only). If not coded, the default is Security level 0 .
		MACRO	PDF417 Macro data. The total length of macro text is limited to 2,710 bytes. This is the maximum number of symbols that can be displayed using PDF417 symbology and all numbers in the data. This does not guarantee that the macro text is not too long if the macro contains other than numeric data, characters, or binary data, for example.
		<u>CHAR</u>	Character format. CHAR is the default.
		HEX	Hexadecimal format.
		<i>macro data</i>	The data that is sent to the printer. If <i>macro data</i> is repeated without a preceding format keyword, it is processed in the same format as the preceding string. Repetition is required for large amounts of data.
	<i>end marker</i>		Always end a command with an end marker (;).

Examples

The following sample program places two differing barcode symbologies:

Draw 2 barcodes in different symbologies:

```

-'**                                     **'
-'**             BCOCA Barcode Overlay           **'
-'**                                     **'
-'**                                     **'
-'**                                     **'
overlay bcocal size 8.5 in 11 in offset 0.25 in 0.25 in ;

define blue1 color oca blue;           -'specify color'

-'**             Barcode definitions             **'
define bc1 barcode code39 1 height 1 in ; -'CODE39 type specifying height'
define bc2 barcode IND2of5 1 color blue1 ; -'Industrial 2 of 5 type, color'

-'**             Barcode placements             **'
position .5 in .5 in;
place barcode bc1 0 '12345';           -'CODE39 at 0 degrees with default HRI'

position right 4 in down 0;
place barcode bc2 90 none '12345';     -'ITL 2 of 5 at 90 degrees with no HRI'

```

Figure 195. Examples for Placing Barcodes

The following sample program displays barcodes of each two-dimensional type:

```

-'*****'
overlay bcg2d1 size 8.4 in 10 in
           offset .2 in .2 in;

-'** Barcode definitions           **'
-'**                               **'
-'** bc4 is for PDF417, modifier 1 **'

define bc1 barcode datamatrix      ;
define bc2 barcode maxicode        ;
define bc3 barcode pdf417          ;
define bc4 barcode pdf417 1        ;

-'**   Sequenced DataMatrix symbols   **'
-'**   **'
-'** The following are 2 symbols that are **'
-'** linked together for the scanner to read **'
-'** together as a single bar code.     **'
-'**   **'
-'** The linkage is via the id parms which **'
-'** must be the same in each symbol portion.**'
-'** The linkage could have been omitted to **'
-'** assume the default of 1 1, but is set **'
-'** to clarify its usage.             **'
-'**   **'
position .3 in .5 in;

```

Figure 196. Examples for Placing Two-dimensional Barcodes (1 of 2)

PLACE BARCODE Command

```
| -'** Structured append 1 of 2          **'|
| place barcode bc1 'DataMatrix Part 1'|
|     2dparms datamatrix rows 10 10    |
|     sequence 1 2 id 1 2              ;|
|                                       |
| position right 2 in down 0;          |
|                                       |
| -'** Structured append 2 of 2          **'|
| place barcode bc1 'DataMatrix Part 2'|
|     2dparms datamatrix rows 10 10    |
|     sequence 2 2 id 1 2              ;|
|                                       |
| -'** Maxicode with defaults except zipper **'|
| -'**                                  **'|
| -'** The following is a symbol that uses **'|
| -'** all the defaults for a Maxicode symbol **'|
| -'** except that a zipper pattern will be **'|
| -'** printed at the right hand side of the **'|
| -'** symbol. This is used as a quick visual **'|
| -'** check for printing distortions.    **'|
| -'**                                  **'|
| position .3 in 3.5 in;               |
| place barcode bc2 'Maxicode with zipper'|
|     2dparms maxicode zipper          ;|
|                                       |
| -'** PDF417 symbols with macros        **'|
| -'**                                  **'|
| -'** The following are 2 symbols that show **'|
| -'** the same symbol data, but the second **'|
| -'** one is shown with type modifier 1. They **'|
| -'** both include a simple macro with the **'|
| -'** data entered in two strings. The rows **'|
| -'** parms ask for the minimum number of **'|
| -'** rows with 8 symbol characters per row. **'|
| -'**                                  **'|
| -'** The modifier can be used in a clean **'|
| -'** environment. It excludes the right row **'|
| -'** indicator keywords, and has a truncated **'|
| -'** stop pattern.                    **'|
| -'**                                  **'|
| position .3 in 6.5 in;               |
| place barcode bc3 'PDF417 with macro'|
|     2dparms pdf417 rows min 8 macro  |
|     '\92812345\111\222\333\222\111\923'|
|     '\000TEST1.TST\923\005106';      |
|                                       |
| position .3 in 7.5 in;               |
| place barcode bc4 'PDF417 with macro'|
|     2dparms pdf417 rows min 8 macro  |
|     '\92812345\111\222\333\222\111\923'|
|     '\000TEST1.TST\923\005106';      |
```

Figure 197. Examples for Placing Two-dimensional Barcodes (2 of 2)

PLACE Command - Segments and Groups

Specify this command if you intend to place a predefined overlay section (**GROUP** or **SEGID**) on the overlay.

Note: You should also issue a **POSITION** command for this overlay section so the program knows where to put it.

PLACE Command — Segments and Groups

▶▶—PLACE—SEGID—*name*—;—▶▶
GROUP

command word

PLACE

section type Specify the type of graphic or overlay section you want placed on this overlay. Choose from:

SEGID Graphic

GROUP Overlay

section name Specify the name of the overlay section you want to place on the overlay. (Use the name that appears on the **GROUP** or **SEGID** command you issued to create the overlay section.)

If you did not assign a name to the segment (in the **SEGID** command), use the *member ID* from that command.

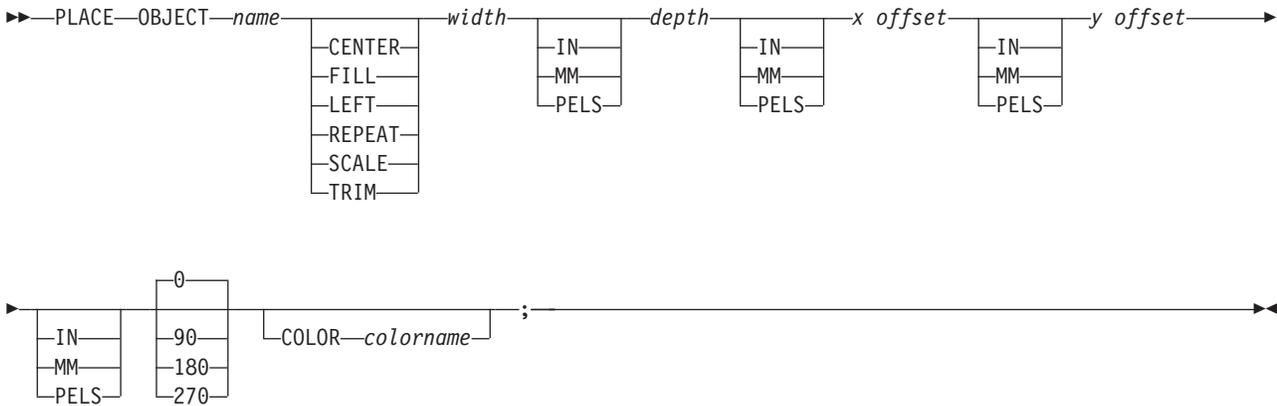
end marker Always end a command with an end marker (;).

PLACE OBJECT Command

PLACE OBJECT Command

Use **PLACE OBJECT** to place an object at the current position on the page.

PLACE — OBJECT Command



name The name of the object definition as specified on the **DEFINE** command.

Position Option

Specify how the object is placed into the object presentation space:

CENTER	center and trim
FILL	scale to fill
LEFT	position
REPEAT	replicate and trim
SCALE	scale to fit
TRIM	position and trim

The positioning value is optional. If not specified, it defaults to the value contained inside the object at print time. For the following object types, **LEFT** is not valid:

- IOCA
- GOCA
- BCOCA
- PSEG

For object type BCOCA only **LEFT** is valid. For object type **OTHER**, mapping options may cause errors at print time.

width Width of the object presentation space. This parameter is required.

depth Depth of the object presentation space. If a unit of measure is not specified, it behaves the same as other commands where units of measure are not specified. This parameter is required.

x offset This is an optional *x* offset inside the object presentation space. See Figure 198 on page 277 for an example.

y offset This is an optional *y* offset inside the object presentation space.

0|90|180|270 Orientation of the object inside the object presentation space. This value is optional.

COLOR Specify the name of a previously-defined color.

Examples:

```
-'Examples of define and place object for AFP Workbench '
CONTROL NOSTORE nosummary;
OVERLAY xxxx SIZE 8.5 IN 11 IN OFFSET 0 0 ;
ORIENT 0;

define xxx object obname tigger obtype other typename tiff;
define yyy object obname schlaf00 obtype pseg;

-'place the ioca pseg into an area twice it's normal size '
-' and bound the area with a box '
position 1 in 1 in;
place object yyy scale 4.8 in .8 in 0 0;
drawbox 4.8 in .8 in;

-'place the tiff object into an area 3 inches by 3.5 inches'
-' and orient at 90 '
position 4 in 4 in;
place object xxx scale 3 in 3.5 in 0 0 90;
position .5 in 4 in;
drawbox 3.5 in 3 in;
```

Figure 198. Examples for Placing Objects through AFP Workbench

```
-'Examples of define and place object to print with PSF/MVS'
CONTROL NOSTORE nosummary;
OVERLAY xxxx SIZE 8.5 IN 11 IN OFFSET 0 0 ;
ORIENT 0;

define yyy object obname schlaf00 obtype pseg;
define zzz object obname bliob3 obtype bcoca;

-'place the ioca pseg into an area twice it's normal size '
-' and bound the area with a box '
position 1 in 1 in;
place object yyy scale 4.8 in .8 in 0 0;
drawbox 4.8 in .8 in;

-'place the bcoca into an area at orient 270 '
-' and bound the area with a box '
position 4 in 4 in;
place object zzz 2 in 1.5 in 0 0 270;
position 4 in 2 in;
drawbox 2 in 1.5 in;
```

Figure 199. Examples for Placing Objects through PSF/MVS

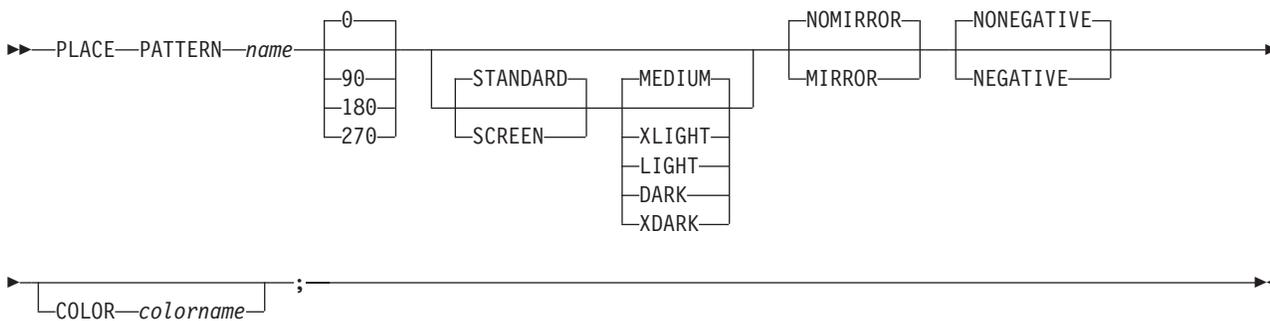
PLACE PATTERN Command

PLACE PATTERN Command

Specify this command if you want to place a pattern on the overlay.

Note: You should also issue a **POSITION** command for this overlay section so the program knows where to put it.

PLACE — PATTERN Command



command word

PLACE

section type

PATTERN

section name Specify the name of the pattern you want to place on the overlay. (Use the name that appears on the **DEFINE PATTERN** command you issued to create this overlay section.)

orientation Specify the orientation of the pattern. Choose from:

0 (Default)

90

180

170

Note: Remember that pattern orientation is relative to the orientation established in the **ORIENT** command.

shading option **SHADE**

shade pattern Choose from:

STANDARD For the pattern shown in Figure 211 on page 326 (default).

SCREEN For the pattern shown in Figure 212 on page 330.

shade type To specify the amount of shading you want for the pattern:

- Enter one of the named shades illustrated in Figure 200 on page 279 that describes the amount of shading you want.
- If you prefer shading that is lighter or darker than one of the named shades, enter a number from 0 to 100 that matches the percentage of shading you want. Do not enter a percent sign, only the number. See Appendix G, "Shade Patterns and Types" on page 325 for an illustration of all percentages. **MEDIUM** is the default.

Note: If the lines of the pattern which you are shading are very fine (1 or 2 pels), it is possible that the shading option you select may make parts of the pattern unclear.

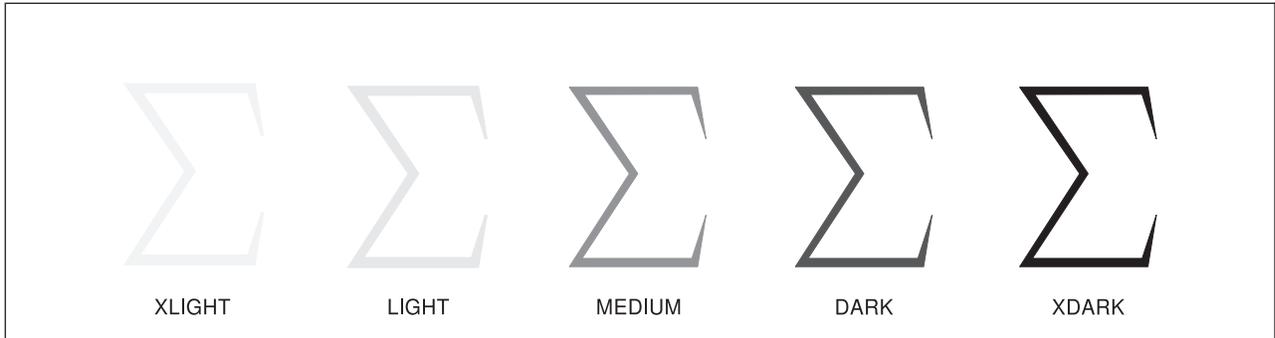


Figure 200. Named Shades for a Pattern

mirror option Choose from:

NOMIRROR Does not print a mirror image of the pattern you are placing (default).
MIRROR Prints a mirror image of the pattern you are placing.

negative option

Choose from:

NONEGATIVE Does not print a negative image of the pattern you are placing (default).
NEGATIVE Prints a negative image of the pattern you are placing.

COLOR

An OCA color definition must be specified. If a color definition other than OCA is specified, an error message is issued and the color request is ignored. Pattern shading is always done with IM1 image.

end marker

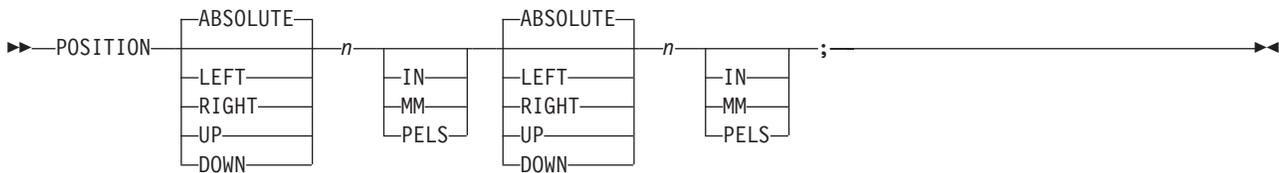
Always end a command with an end marker (;).

POSITION Command

POSITION Command

Use this command to establish a position on the overlay. This position specifies where an object is to be placed. See “Top-Left and Center Positioning with **SETUNITS**” on page 144 for a description of how boxes are placed. See “Positioning the Path (**POSITION**)” on page 59 for a description of how paths and rules are placed. See “Positioning the Circle (**POSITION**)” on page 55 for a description of how circles are placed.

POSITION Command



command word

POSITION

origin option

Specify which kind of positioning to use for the first coordinate:

Absolute positioning: The first coordinate is measured from the overlay (or **GROUP**) origin; enter **ABSOLUTE** (default).

Relative positioning: The first coordinate is measured from the previous point on the path; enter **LEFT**, **RIGHT**, **UP**, or **DOWN**.

first coordinate

Using a number (*n*) and a unit of measurement, specify the first of two coordinates necessary to position the element. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

Note: If you do not specify a first coordinate, the program defaults to the current value in the **SETUNITS** command.

origin option

Specify which kind of positioning to use for the second coordinate:

Absolute positioning: The second coordinate is measured from the overlay (or **GROUP**) origin; enter **ABSOLUTE** (default).

Relative positioning: The second coordinate is measured from the previous point on the path; enter **LEFT**, **RIGHT**, **UP**, or **DOWN**.

second coordinate

Using a number (*n*) and a unit of measurement, specify the first of two coordinates necessary to position the element. Choose from:

n

IN Inches
MM Millimeters
PELS Pels

Note: If you do not specify a second coordinate, the program defaults to the current value in the **SETUNITS** command.

end marker

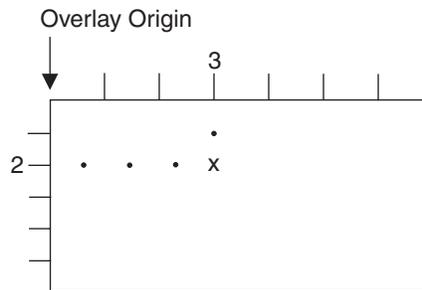
Always end a command with an end marker (;).

Note:

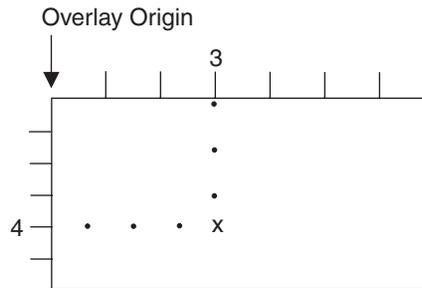
- You can mix **ABSOLUTE** with **RIGHT**, **LEFT**, **DOWN**, or **UP** in the same **POSITION** command. However, when **ABSOLUTE** is specified as the *first coordinate*, it is the horizontal coordinate. Specified as the *second coordinate*, **ABSOLUTE** is the vertical coordinate.
- You can mix any two keywords (**LEFT** and **RIGHT** or **UP** and **DOWN**, for example) in the same **POSITION** command, or you can use the same keyword twice (**DOWN** and **DOWN**).
- You must enter two coordinates, even if one is a 0 coordinate. If you are making a relative move and you want to move only in one direction, you still must specify the 0 direction.

The example below shows what can happen if you do not specify the 0 direction:

The current position is the coordinate (3,2).

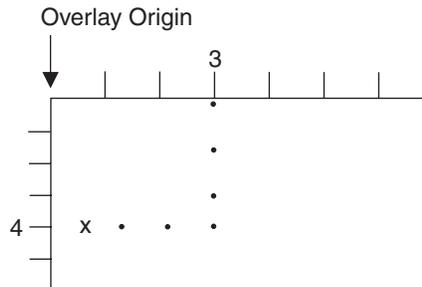


The desired position is the coordinate (3,4).



POSITION Command

To get there from coordinate (3,2), you should specify **POSITION DOWN 2 RIGHT 0**. However, If you specify **POSITION DOWN 2 0**, what you get is:

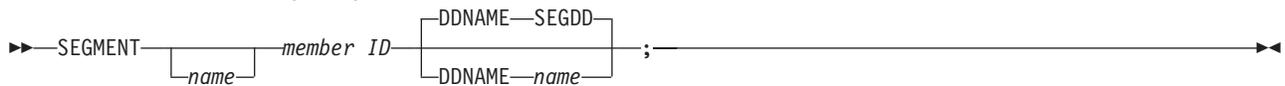


Because you did not specify the 0 direction, OGL/370 assumes that the distance should be measured from the overlay origin, rather than from the current position as you intended.

SEGMENT Command (MVS)

In an MVS environment, use this command to include a page segment in your overlay. You can include as many segments as you wish in your overlay, but remember to reissue the entire command for each segment you include.

SEGMENT Command (MVS)



command word

SEGMENT

segment name

An optional entry, a name used to place the segment. If you do not make a segment name entry, the program uses the *member ID* (the next entry in this command) as the segment name in this command, and you must use that *member ID* name in the **PLACE** command for this segment.

The segment name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

Ask your system programmer for the member name under which the segment is stored in the library.

member ID

Specify the identifier of the segment you want to use.

The *member ID*, which cannot be more than six characters long, is derived from the member name under which the segment is stored in the segment library. To derive the *member ID*, remove the two-character prefix from the member name. For example, the segment for the palm tree logo in Figure 14 on page 23 is stored in the segment library under the member name "S1PALM2". To get this segment in your overlay, you would specify "PALM2" as your *member ID* entry. If you orient the overlay, make sure you have the appropriate segment for that orientation. Remember that a segment only prints in the orientation in which it was designed. For more information see "Placing Graphics (**PLACE**)" on page 125.

DD statement word and statement name

Specify the **DDNAME** name, if the *DD statement name* is something other than **SEGDD**. The *DD statement name* is specified in the JCL. See page "**SEGMENT Command (MVS)**" for an example of an MVS **SEGMENT** command. If the name is no **SEGDD**, use the first choice below:

DDNAME name

Enter **DDNAME name**, to specify a DD statement name other than **SEGDD**.

DDNAME SEGDD

(Default)

end marker

Always end a command with an end marker (;).

SEGMENT Command (VM)

SEGMENT Command (VM)

In a VM environment, use this command to include a page segment in your overlay. You can include as many segments as you wish in your overlay, but remember to reissue the entire command for each segment you include.

SEGMENT Command (VM)



command word

SEGMENT

segment name

An optional entry, segment name is a 1- to 6-character name you can use to place the segment. If you do not make a segment name entry, the program uses the *member ID* (the next entry in this command) as the segment name in this command, and you must use that name in the **PLACE** command for this segment.

The segment name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

Ask your system programmer for the name under which the segment is stored.

member ID

Specify the identifier of the segment you want to use.

The *member ID*, which cannot be more than six characters long, is derived from the filename under which the segment is stored. To derive the *member ID*, remove the two-character prefix from the filename. For example, the segment for the palm tree logo in Figure 14 on page 23 is stored in the segment library under the filename "S1PALM2". To get this segment in your overlay, you would specify "PALM2" as your *member ID* entry. If you orient the overlay, make sure you have the appropriate segment for that orientation. Remember that a segment will print only in the orientation in which it was designed. For more information, see "Placing Graphics (**PLACE**)" on page 125.

filetype indicator and filetype name

Specify **FILETYPE** name, if the filetype is something other than **PSEG38PP**. The filetype name is specified in the JCL. If the name is not **PSEG38PP**, use the first choice below:

FILETYPE *name*

Enter **FILETYPE** *name*, to specify a filetype indicator other than **PSEG38PP**.

FILETYPE PSEG38PP

(Default)

end marker

Always end a command with an end marker (;).

SEGMENT Command (VSE)

In a VSE environment, use this command to include a page segment in your overlay. You can include as many segments as you wish in your overlay, but remember to reissue the entire command for each segment you include.

SEGMENT Command (VSE)

```
▶—SEGMENT—name—member ID—;—▶
```

command word

SEGMENT

segment name

An optional entry, a name used to place the segment. If you do not make a segment name entry, the program uses the segment ID (the next entry in this command) as the segment name in this command, and you must use that name in the **PLACE** command for this segment.

The segment name must meet these requirements:

- The name can include only the following characters: A-Z, 0-9, @, #, -, and \$.

Note: The first character cannot be a hyphen (-).

- The name cannot be longer than six characters.

Ask your system programmer for the name under which the segment is stored in the library.

member ID

Specify the identifier of the segment you want to use.

The *member ID*, which cannot be more than six characters long, is derived from the segment name under which the segment is stored in the segment library. To derive the *member ID*, remove the two-character prefix from the segment name. For example, the segment for the palm tree logo in Figure 14 on page 23 is stored in the segment library under the segment name “S1PALM2”. To get this segment in your overlay, you would specify “1PALM2” as your *member ID* entry. For more information see “Placing Graphics (**PLACE**)” on page 125.

end marker

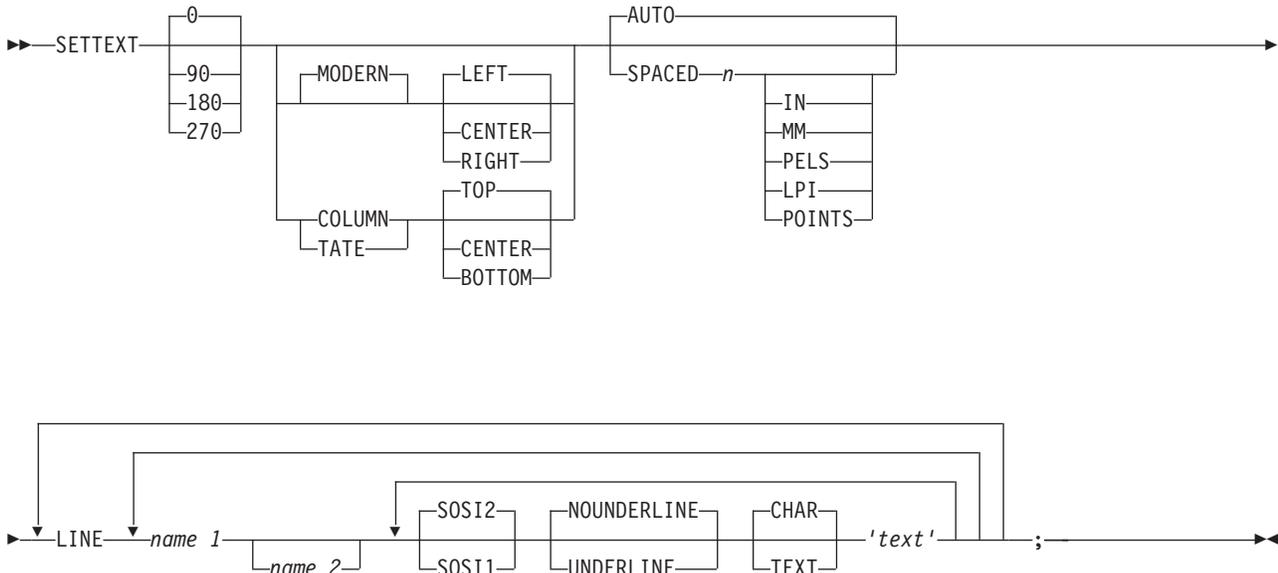
Always end a command with an end marker (-).

SETTEXT Command

SETTEXT Command

Issue this command to place text in the overlay. The positioning of text with **SETTEXT** is described in detail under “Adding Text Another Way (**SETTEXT**)” on page 108.

SETTEXT Command



command word

SETTEXT

orientation

Specify the orientation of the text that is to be placed in the overlay. Choose from:

0 (Default)

90

180

270

Notes:

1. Remember that the orientation you specify is relative to the overlay.
2. Before you specify orientation and format (next entry), make sure the font you select exists for that combination.

format

Select the format of the text you want placed in the overlay. Choose from:

MODERN

Characters are printed from left to right (default). As the sentences appear on the pages of this manual.

COLUMN

Characters are printed vertically from top to bottom, and text strings are printed from left to right. If you specify **COLUMN** with two text strings, it would have the following result:

```

T   L
w   i
o   n
    e
    s
  
```

TATE

Characters are printed vertically from top to bottom, and text strings are printed from right to left, like this:

L T
i w
n o
e
s

alignment Indicate the alignment of text within the overlay.

Note: Remember to specify the text origin in the **POSITION** command that appears just before this **SETTEXT** command. Refer to “Adding Text Another Way (**SETTEXT**)” on page 108 to determine the proper text origin entry for your text.

As you can see in the **SETTEXT** syntax diagram, the text alignment entries for **MODERN** (default), format are different from those for the **COLUMN** and **TATE** formats.

If you select **MODERN** format, you can specify three different text alignments. See Figure 201. Choose from:

- LEFT** Each text string begins directly under the beginning of the first text string (default).
- BOTTOM** The center of each text string is directly under the center of the first text string.
- CENTER** Each text string ends directly under the end of the first text string.

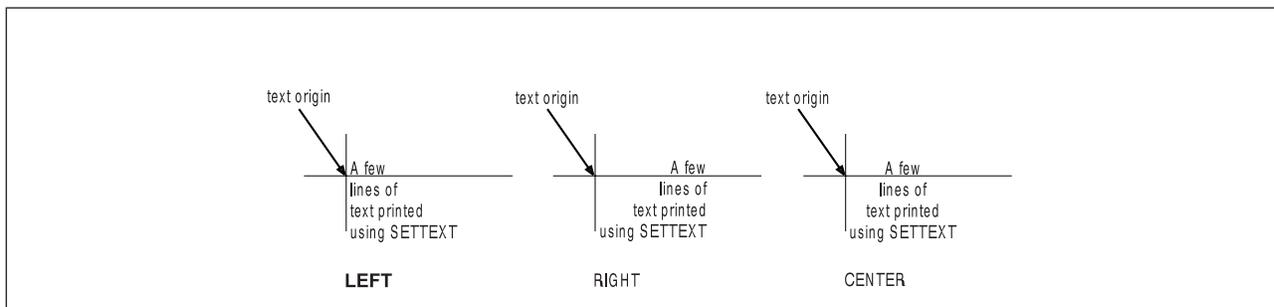


Figure 201. Alignment Options Using the **MODERN** Format.

If you specify either **COLUMN** or **TATE** format, you can also choose from among three different text alignments within the overlay. See Figure 202 on page 288 and Figure 203 on page 288. Choose from:

- TOP** Begins each text string to the right (**COLUMN**) or left (**TATE**) of the beginning of the first text string (default).
- BOTTOM** Ends each text string directly to the right (**COLUMN**) or left (**TATE**) of the end of the first text string.
- CENTER** Aligns the center of each text string directly to the right (**COLUMN**) or left (**TATE**) of the center of the first text string.

SETTEXT Command

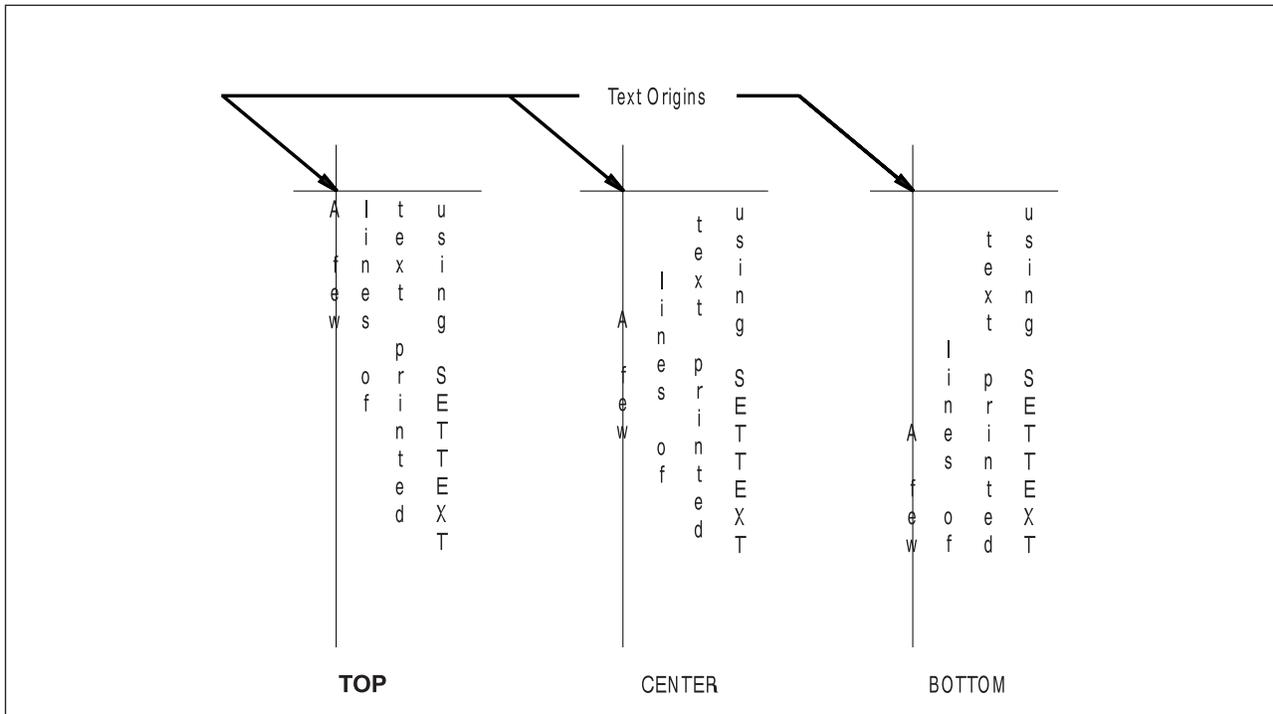


Figure 202. Alignment Options Using the **COLUMN** Format

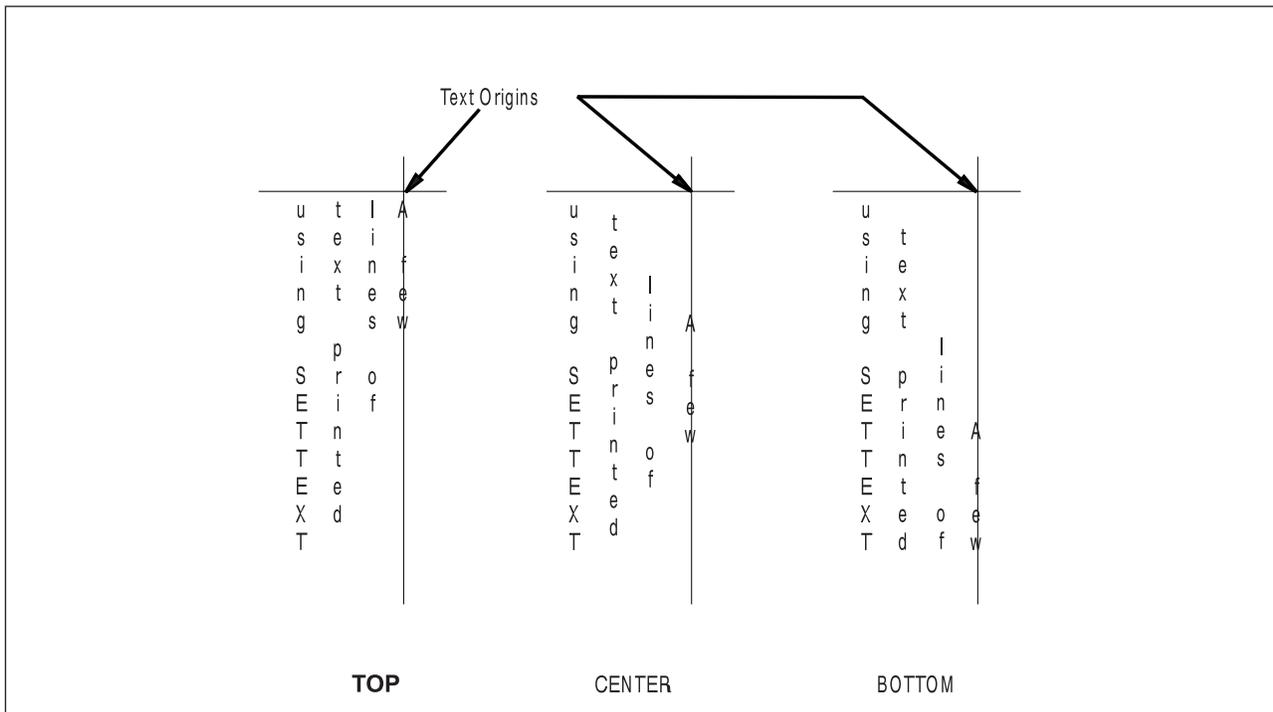


Figure 203. Alignment Options Using the **TATE** Format

line spacing If there are two or more text strings in the text block, enter the type of line spacing you want to use:

AUTO Calculates line spacing based on the fonts used for the text (default).

SPACED Indicates that you intend to specify the spacing between text strings. To do this, enter a number and a unit of measurement for the distance you want between the baselines of text.

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch

Note: You can also default to the current **LINESP** value established in the **SETUNITS** command.

subcommand word

To begin defining a text string, you must enter the subcommand word:

LINE

You must enter the word **LINE** for each text string.

font names

Enter the names of the fonts you want to use for the text strings that follow. Specify either one or two fonts. One font is used for SBCS characters and the other is used for DBCS characters. The SBCS font specified is used for all SBCS characters in all subsequent text strings in the **LINE** subcommand until another SBCS font is designated. Similarly, the DBCS font specified remains in effect until another DBCS font is designated.

Note: The optional second font name is ignored when you specify the **NOSOSI** option in the **CONTROL** command.

If a text string contains any SBCS characters, you need to specify an SBCS font. If a text string contains any DBCS characters, you need to specify a DBCS font. If a text segment contains both SBCS and DBCS characters, you need to specify two fonts—one SBCS font and one DBCS font, the order is irrelevant.

If you use DBCS text and specify **SOSI1** mode, the SOSI delimiters appear as SBCS spaces. If you use **SOSI1** mode, you must specify an SBCS font in the **LINE** subcommand before you specify any text strings containing DBCS characters.

Each font name specified in the **LINE** subcommand must already be named in a previous **FONT** command.

To change fonts in a text string:

1. Complete the remaining **SETTEXT** command entries (underlining, text type, and text) for the text you want in the original font.
2. Enter the name of the new font or fonts you want to use and complete the remaining entries for that text string.

SOSI mode

This option defines the way SOSI delimiters are to be handled. This option is ignored if **NOSOSI** is specified in the **CONTROL** command.

SOSI1 A single-byte character space appears wherever SOSI delimiters occur.

When a SO delimiter follows single-byte text, the font used to determine the size of the character space is the same font used for the preceding single-byte text. When a SI delimiter is followed by single-byte text, the font used to determine the size of the character space is the same font used for the following single-byte text. If no single-byte text either precedes a SO delimiter or follows a SI delimiter, the last single-byte font specified is used to determine the size of the character spaces.

SETTEXT Command

- SOSI2** No character space appears in the positions held by SOSI delimiters (default).
- underlining* Indicates whether you want the following text segment underlined. Blank spaces included as part of the text segment, are also underlined. Choose from:
NOUNDERLINE Does not underline the following text segment (default).
UNDERLINE Underlines the following text segment.
- Notes:**
1. Only text written in **MODERN** format can be underlined.
 2. You must specify each text segment you want underlined.
- text type* Specify text type as follows:
CHAR The characters you type are the exact characters that are to be printed on the overlay (default).
HEX The text string to follow is in hexadecimal form.
- Notes:**
1. Refer to page 112 for information on hex text.
 2. For information about double-byte fonts, see “Symbolic Data Sets and Symbolic Files Containing Double-Byte Characters” on page 320.
- text* Enter the text you want placed in the overlay. Each part of a line that uses different characteristics (**FONT**, **UNDERLINE/NOUNDERLINE**, **CHAR/HEX**) must be enclosed in apostrophes as a unit.
- Blanks in Balanced Text**
- If you specified **BALANCE** for the text placement, you should not include blanks between the apostrophes. However, if you want to include blanks in text that is to be balanced, you can do it by specifying multiple text segments: one segment for each character string (excluding blanks) and one segment for each string of blanks. For an example of how this balancing feature works, see “Blanks in Balanced Text” on page 96.
- Symbolic Names**
- If there is a text string that you wish to use on several different overlays, you can give a symbolic name to that text string. Each time you include the symbolic name in a **SETTEXT** command, the text string represented by that symbolic name, is placed on the overlay.
- See “Symbolic Data Sets or Files” on page 114 for more information.
- Notes:**
1. For single-byte fonts, precede a symbolic name with an ampersand (&) and follow it with a period.
 2. For double-byte fonts, the symbolic name should begin with “&#”, and terminate with a period.
- end marker* Always end a command with an end marker (-).

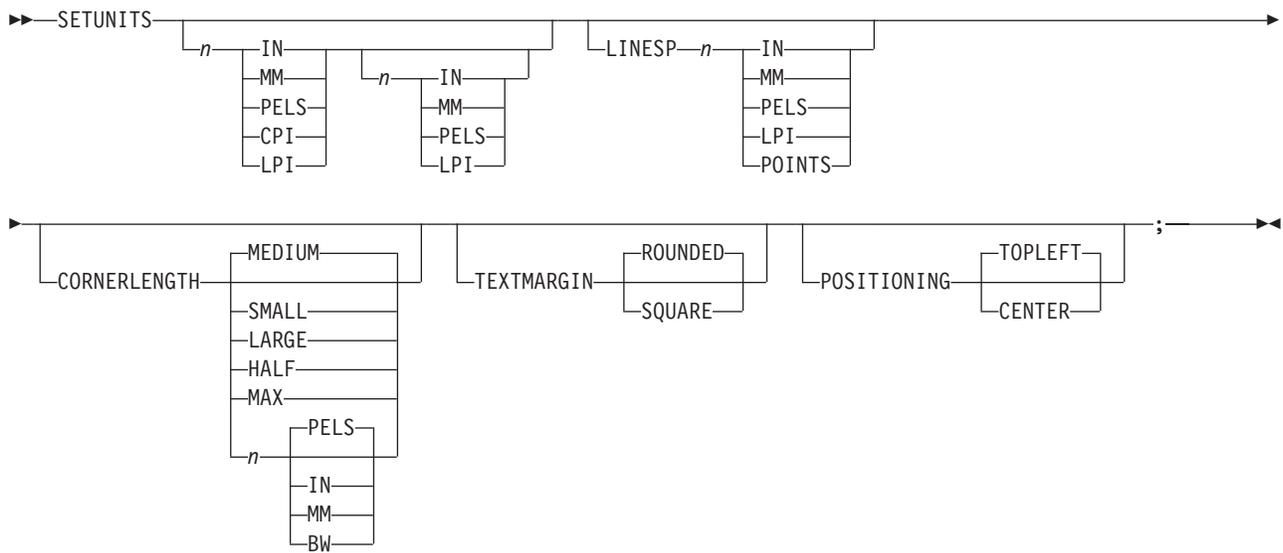
SETUNITS Command

Use this command to define the following:

- Default unit of measurement
- Default line spacing
- Corner length for rounded box corners and path connections
- Text margin to be used when placing text in boxes and circles
- Positioning method to be used for boxes and rules.

In general, any option you set with **SETUNITS** stays in force until you change it by specifying it again in another **SETUNITS** command.

SETUNITS Command



command word

SETUNITS

primary default

With a number (n) and a unit of measurement, specify the default unit of measurement for horizontal distance and spacing. Choose from:

- n
- IN** Inches
 - MM** Millimeters
 - PELS** Pels
 - CPI** Characters per inch
 - LPI** Lines per inch

Note: If you specify only the primary default, the unit of measurement you specify is also used as the default for the vertical distance and spacing.

SETUNITS Command

secondary default

With a number (*n*) and unit of measurement, specify the default unit of measurement for vertical distance and spacing. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch

linespace option

Enter **LINESP** to indicate that you want to establish a default value for use in the **SPACED** option of the following commands:

- **DRAWBOX (WITHTEXT)**
- **DRAWCIRCLE (WITHTEXT)**
- **SETTEXT**

linespace value

With a number (*n*) and a unit of measurement, specify the default value for the **SPACED** option of the commands listed above. Choose from:

n

IN	Inches
MM	Millimeters
PELS	Pels
LPI	Lines per inch
POINTS	Points

Notes for Figure 204

1. If you specify only **LINESP** in a second **SETUNITS** command, the previously established primary and secondary distance and spacing defaults, if any, remain in effect.
2. If you do not specify **LINESP** in a second **SETUNITS** command and provide only primary and secondary distance and spacing defaults, the previous linespace values, if any, remain in effect.
3. If **LINESP** is not specified, the default unit of measurement for line spacing is either the primary or secondary default:
 - If you specify only a primary default, it applies to both horizontal and vertical text strings.
 - If you specify primary and secondary defaults, the primary default applies to line spacing for lines of text that are vertical relative to the overlay. The secondary default applies to line spacing for lines of text that are horizontal relative to the overlay. Sample A in Figure 204 uses the primary default as a line spacing unit of measurement; Sample B uses the secondary default.

Sample A

V l t - t
e i e t h
r n x i e
t e t v
i s e o
c r v
a o e t e
l f l o r

Sample B

Horizontal lines
of text relative
to the overlay

Figure 204. Choosing the Line-Spacing Default

subcommand word

Specify **CORNERLENGTH** if you want to define the corner length for **DRAWBOX** corners and default corner length for **DRAWPATH** rounded connections.

length value

Specify the length of the rounded corners for the **DRAWBOX** command and the default length of rounded connections for the **DRAWPATH** command. Choose from:

- SMALL**
- MEDIUM** (Default)
- LARGE**
- HALF**
- MAX**

For boxes, **MAX** and **HALF** give a rounded corner with a length $\frac{1}{2}$ the length of the shortest side of the box.

For paths, **MAX** gives an arc that extends the full length of the shorter of the two segments being connected. **HALF** gives an arc that extends $\frac{1}{2}$ that length.

Instead of a keyword, you can enter a value (*n*) and unit of measurement:

n

- IN** Inches
- MM** Millimeters
- PELS** Pels (default)
- BW** Multiples of the border weight

Note: If you specify a number with no unit of measurement, the program uses **PELS**.

subcommand word

Specify **TEXTMARGIN** if you want to select a text margin type.

text margin

Specify the text margin type you require. **ROUNDED** selects a margin that is exactly one border thickness inside the border of the box or circle all the way around. **SQUARE** is a rectangle inside the **ROUNDED** margin. It meets the **ROUNDED** margin at rounded corners where the 45 degree line bisects the corner. Note that a square margin is identical to a **ROUNDED** margin in a square-cornered box. See "Text Margins in Boxes" on page 70 and "Text Margins in Circles" on page 73 for an explanation of text margins.

subcommand word

Specify **POSITIONING** if you want to select a type of positioning for boxes and rules.

positioning option

Specify the kind of positioning you require for boxes and rules.

- TOPLEFT** The top-left corner of the box or rule is placed at the position you specified in the last **POSITION** command.
- CENTER** The center of the border at the top-left corner of the box or rule is placed at that position.

Refer to "Top-Left and Center Positioning with **SETUNITS**" on page 144 for a detailed description of top-left and center positioning. Figure 205 on page 294 shows the result of choosing each positioning option for boxes.

SETUNITS Command

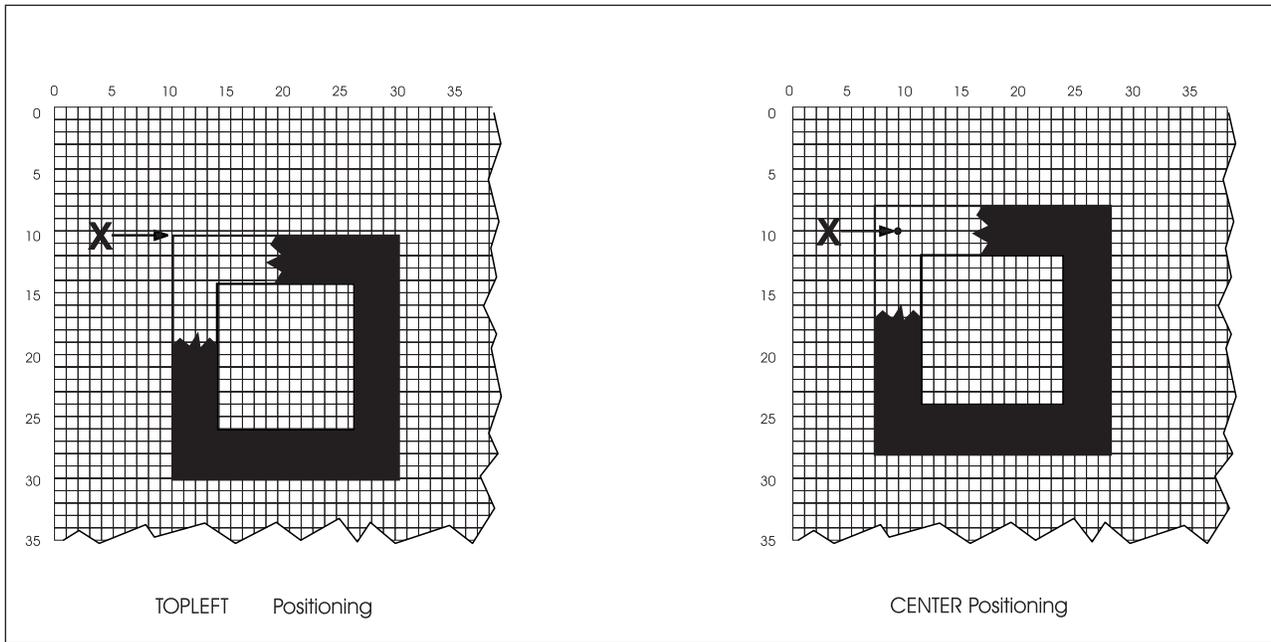


Figure 205. **TOPLEFT** and **CENTER** Positioning for Boxes

Note: With any of these options (corner length, text margin and box positioning), once they are specified in a **SETUNITS** command, the value you specify will remain in effect until the option is re-specified in another **SETUNITS** command. For example, if you specify the **CORNERLENGTH** option and the **POSITIONING** option in a **SETUNITS** command, these become the current values for these options. If, in a second **SETUNITS** command, you specify only the **POSITIONING** option, this becomes the current value. The current value for a corner length is the value specified in the first **SETUNITS** command.

end marker Always end a command with an end marker (;).

LINE NUM.	S O U R C E I N P U T S T A T E M E N T S	SEQUENCE -NUMBER-
	-----1-----2-----3-----4-----5-----6-----7--	
0034	POSITION ABSOLUTE 1 IN ABSOLUTE 1.05 IN;	00340020
0035	DRAWRULE ACROSS .75 IN LIGHT SOLID;	00350020
0036	POSITION ABSOLUTE 1 IN ABSOLUTE 1.3 IN;	00360020
0037	DRAWRULE ACROSS .75 IN LIGHT SOLID;	00370020
0038	POSITION ABSOLUTE 1.45 IN ABSOLUTE .55 IN;	00380020
0039	DRAWRULE DOWN 1 IN LIGHT DASHED;	00390020
0040		00400020
0041	POSITION ABSOLUTE 3.25 IN ABSOLUTE .75 IN; -'NAME LINE'	00410020
0042	DRAWRULE ACROSS 2.5 IN MEDIUM SOLID;	00420020
0043		00430020
0044	POSITION ABSOLUTE 3.25 IN ABSOLUTE 1.25 IN;	00440020
0045		00450020
0046	DRAWRULE ACROSS 2.5 IN MEDIUM SOLID;	00460020
0047	POSITION ABSOLUTE 4 IN ABSOLUTE 1.1 IN;	00470020
0048	DRAWRULE ACROSS .15 IN MEDIUM SOLID;	00480020
0049	POSITION ABSOLUTE 4.8 IN ABSOLUTE 1.1 IN;	00490020
0050	DRAWRULE ACROSS .15 IN MEDIUM SOLID;	00500020
0051		00510020
0052	POSITION ABSOLUTE 6.1 IN ABSOLUTE .4 IN; -'NOTICE LINE'	00520020
0053	DRAWRULE DOWN 2.6 IN MEDIUM SOLID;	00530020
0054		00540020
0055	POSITION ABSOLUTE 3.25 IN ABSOLUTE 2 IN;	00550020
0056		00560020
0057	DRAWRULE ACROSS 1.75 IN MEDIUM SOLID;	00570020
0058		00580020
0059	POSITION ABSOLUTE 5.25 IN ABSOLUTE 2 IN; -'DATE LINE'	00590020
0060	DRAWRULE ACROSS .5 IN MEDIUM SOLID;	00600020
0061		00610020
0062	POSITION ABSOLUTE 3.25 IN ABSOLUTE 2.5 IN;	00620020
0063		00630020
0064	DRAWRULE ACROSS 2.5 IN MEDIUM SOLID;	00640020
0065		00650020
0066		00660020
0067	-'DRAWING BOXES'	00670020
0068	POSITION ABSOLUTE 0 ABSOLUTE 0; -'RECEIPT FRAME'	00680020
0069	DRAWBOX 7 IN 3 IN MEDIUM SOLID;	00690020
0070		00700020
0071	POSITION ABSOLUTE .25 IN ABSOLUTE .55 IN; -'UNITS BOX'	00710020
0072	DRAWBOX .37 IN .37 IN BOLD SOLID;	00720020
0073		00730020
0074	POSITION ABSOLUTE 1 IN ABSOLUTE .55 IN; -'FEES BOX'	00740020
0075	DRAWBOX .75 IN 1 IN MEDIUM SOLID SHADE LIGHT;	00750020
0076		00760020

LINE NUM.	S O U R C E I N P U T S T A T E M E N T S	SEQUENCE -NUMBER-
	-----1-----2-----3-----4-----5-----6-----7--	
0077	POSITION ABSOLUTE 3.25 IN ABSOLUTE 1.75 IN;	00770020
0078		00780020
0079	DRAWBOX 1.75 IN .25 IN 0 SHADE LIGHT;	00790020
0080		00800020
0081		00810020
0082	- 'ADDING WORDS'	00820020
0083		00830020
0084	POSITION ABSOLUTE 6.2 IN ABSOLUTE .5 IN; -NOTICE	00840020
0085	DRAWBOX .7 IN 2.4 IN BOLD DASHED	00850020
0086	WITHTEXT 270 MODERN TOP CENTER SPACED .2 IN	00860020
0087	LINE FONT3 NOUNDERLINE CHAR 'Keep this copy.'	00870020
0088	LINE FONT3 NOUNDERLINE CHAR 'NOT VALID as ID.'	00880020
0089	WITHTEXT 0 COLUMN RIGHT BALANCE	00890020
0090	LINE FONT2 NOUNDERLINE CHAR 'NOTICE';	00900020
0091		00910020
0092		00920020
0093	POSITION ABSOLUTE 0 ABSOLUTE 0; -HEADER	00930020
0094	DRAWBOX 7 IN .4 IN 0 SHADE LIGHT	00940020
0095	WITHTEXT 0 MODERN CENTER CENTER	00950020
0096	LINE FONT1 NOUNDERLINE CHAR 'DO '	00960020
0097	FONT2 NOUNDERLINE CHAR 'NOT '	00970020
0098	FONT1 NOUNDERLINE CHAR 'WRITE IN AREA BELOW';	00980020
0099		00990020
0100		01000020
0101	POSITION ABSOLUTE .25 IN ABSOLUTE 1.1 IN; -UNITS	01010020
0102	DRAWBOX .5 IN .5 IN 0	01020020
0103	WITHTEXT 0 MODERN TOP LEFT SPACED .15 IN	01030020
0104	LINE FONT5 NOUNDERLINE CHAR 'Total'	01040020
0105	LINE FONT5 NOUNDERLINE CHAR 'Units';	01050020
0106		01060020
0107	POSITION ABSOLUTE .75 IN ABSOLUTE .55 IN; -FEES	01070020
0108	DRAWBOX .25 IN 1 IN 0 SOLID	01080020
0109	WITHTEXT 90 MODERN CENTER BALANCE	01090020
0110	LINE FONT3 NOUNDERLINE CHAR 'FEES';	01100020
0111	POSITION ABSOLUTE 1.9 IN ABSOLUTE .5 IN;	01110020
0112	DRAWBOX 1 IN 1 IN 0	01120020
0113	WITHTEXT 0 MODERN BOTTOM LEFT SPACED .25 IN	01130020
0114	LINE FONT5 NOUNDERLINE CHAR 'Registration'	01140020
0115	LINE FONT5 NOUNDERLINE CHAR 'Out-of-State'	01150020
0116	LINE FONT5 NOUNDERLINE CHAR 'Other'	01160020
0117	LINE FONT5 UNDERLINE CHAR 'Total';	01170020
0118		01180020
0119	POSITION ABSOLUTE 3.25 IN ABSOLUTE .75 IN; -NAME	01190020

LINE NUM.	SOURCE INPUT STATEMENTS	SEQUENCE -NUMBER-
	-----1-----2-----3-----4-----5-----6-----7--	
0120	DRAWBOX 2.5 IN .2 IN 0	01200020
0121	WITHTEXT 0 MODERN BOTTOM LEFT	01210020
0122	LINE FONT5 NOUNDERLINE CHAR	01220020
0123	'LAST NAME FIRST NAME MI';	01230020
0124		01240020
0125	POSITION ABSOLUTE 3.25 IN ABSOLUTE 1.25 IN;	01250020
0126		-'SOC. SEC. NO.'
0127	DRAWBOX 2.5 IN .2 IN 0	01270020
0128	WITHTEXT 0 MODERN BOTTOM CENTER	01280020
0129	LINE FONT5 NOUNDERLINE CHAR 'SOCIAL SECURITY NUMBER';	01290020
0130		01300020
0131	POSITION ABSOLUTE 3.25 IN ABSOLUTE 2 IN;	-RECEIVED
0132	DRAWBOX 1 IN .25 IN 0	01320020
0133	WITHTEXT 0 MODERN LEFT CENTER	01330020
0134	LINE FONT5 NOUNDERLINE CHAR 'Received by';	01340020
0135		01350020
0136	POSITION ABSOLUTE 5.25 IN ABSOLUTE 2 IN;	-DATE
0137	DRAWBOX .5 IN .25 IN 0	01370020
0138	WITHTEXT 0 MODERN LEFT CENTER	01380020
0139	LINE FONT5 NOUNDERLINE CHAR 'Date';	01390020
0140		01400020
0141	POSITION ABSOLUTE 1.25 IN ABSOLUTE 1.75 IN;	
0142		-COLLEGE
0143	DRAWBOX 1.75 IN .6 IN 0	01430020
0144	WITHTEXT 0 MODERN CENTER CENTER SPACED .2 IN	01440020
0145	LINE FONT2 NOUNDERLINE CHAR 'TropiCal'	01450020
0146	LINE FONT2 NOUNDERLINE CHAR 'Community College';	01460020
0147		01470020
0148	POSITION ABSOLUTE 3.25 IN ABSOLUTE 2.65 IN;	-PRESIDENT
0149	DRAWBOX 1.25 IN .5 IN 0	01490020
0150	WITHTEXT 0 MODERN LEFT TOP SPACED .2 IN	01500020
0151	LINE FONT5 NOUNDERLINE CHAR 'Robinson K. Russo'	01510020
0152	LINE FONT5 NOUNDERLINE CHAR 'President';	01520020
0153		01530020
0154	-'INITIALS PATTERN'	01540020
0155	DEFINE BIGT PATTERN ENCODED (0 80) -1	01550020
0156		(0 80) -2
0157		(0 80) -3
0158		(0 80) -4
0159		(0 80) -5
0160		(0 80) -6
0161		(0 80) -7
0162		(0 80) -8

LINE NUM.	S O U R C E	I N P U T	S T A T E M E N T S	SEQUENCE -NUMBER-
	-----1-----	-----2-----	-----3-----4-----5-----6-----7--	
0163			(0 80) -9	01630020
0164			(0 80) -10	01640020
0165			(0 80) -11	01650020
0166			(0 80) -12	01660020
0167			(0 80) -13	01670020
0168			(0 80) -14	01680020
0169			(0 80) -15	01690020
0170			(0 80) -16	01700020
0171			(0 80) -17	01710020
0172			(0 80) -18	01720020
0173			(0 80) -19	01730020
0174			(0 80) -20	01740020
0175			(0 80) -21	01750020
0176			(0 80) -22	01770020
0177			(0 80) -23	01780020
0178			(0 80) -24	01790020
0179			(0 80) -25	01800020
0180			(0 80) -26	01810023
0181			(27 26) -27	01820020
0182			(27 26) -28	01830020
0183			(27 26) -29	01840020
0184			(27 26) -30	01850020
0185			(27 26) -31	01860020
0186			(27 26) -32	01870020
0187			(27 26) -33	01880020
0188			(27 26) -34	01890020
0189			(27 26) -35	01900020
0190			(27 26) -36	01910020
0191			(27 26) -37	01920020
0192			(27 26) -38	01930020
0193			(27 26) -39	01940020
0194			(27 26) -40	01950020
0195			(27 26) -41	01960020
0196			(27 26) -42	01970020
0197			(27 26) -43	01980020
0198			(27 26) -44	01990020
0199			(27 26) -45	02000020
0200			(27 26) -46	02010020
0201			(27 26) -47	02020020
0202			(27 26) -48	02030020
0203			(27 26) -49	02040020
0204			(27 26) -50	02050020
0205			(27 26) -51	02060020

LINE NUM.	S O U R C E	I N P U T	S T A T E M E N T S	SEQUENCE -NUMBER-				
	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7--	
0206			(27 26)	-52				02070020
0207			(27 26)	-53				02080020
0208			(27 26)	-54				02090020
0209			(27 26)	-55				02100020
0210			(27 26)	-56				02110020
0211			(27 26)	-57				02120020
0212			(27 26)	-58				02130020
0213			(27 26)	-59				02140020
0214			(27 26)	-60				02150020
0215			(27 26)	-61				02160020
0216			(27 26)	-62				02170020
0217			(27 26)	-63				02180020
0218			(27 26)	-64				02190020
0219			(27 26)	-65				02200020
0220			(27 26)	-66				02210020
0221			(27 26)	-67				02220020
0222			(27 26)	-68				02230020
0223			(27 26)	-69				02240020
0224			(27 26)	-70				02250020
0225			(27 26)	-71				02260020
0226			(27 26)	-72				02270020
0227			(27 26)	-73				02280020
0228			(27 26)	-74				02290020
0229			(27 26)	-75				02300020
0230			(27 26)	-76				02310020
0231			(27 26)	-77				02320020
0232			(27 26)	-78				02330020
0233			(27 26)	-79				02340020
0234			(27 26)	-80;				02350020

DZI0415I PATTERN - INFORMATIONAL MESSAGE:
 THE PATTERN "BIGT", AS SPECIFIED, IS 80 PELS WIDE BY 80 PELS
 HIGH. THE PATTERN SIZE (A MULTIPLE OF EIGHT PELS) IS 80 PELS
 WIDE BY 80 PELS HIGH.

0235								02360020
0236	DEFINE SMALLC PATTERN ENCODED		(17 6)	-1				02370020
0237			(14 12)	-2				02380020
0238			(12 16)	-3				02390020
0239			(10 20)	-4				02400020
0240			(9 22)	-5				02410020
0241			(7 26)	-6				02420020
0242			(6 28)	-7				02430020

LINE NUM.	S O U R C E	I N P U T	S T A T E M E N T S	SEQUENCE -NUMBER-
	-----1-----	-----2-----	-----3-----4-----5-----6-----7--	
0243			(5 30) -8	02440020
0244			(5 30) -9	02450020
0245			(4 32) -10	02460020
0246			(3 15 4 15) -11	02470020
0247			(3 12 10 12) -12	02480020
0248			(2 12 12 12) -13	02490020
0249			(2 11 14 13) -14	02500020
0250			(2 10 16 12) -15	02510020
0251			(1 11 16 12) -16	02520020
0252			(1 11) -17	02530020
0253			(0 11) -18	02540020
0254			(0 11) -19	02550020
0255			(0 11) -20	02560020
0256			(0 11) -21	02570020
0257			(0 11) -22	02580020
0258			(0 11) -23	02590020
0259			(1 11) -24	02600020
0260			(1 11 16 12) -25	02610020
0261			(2 10 16 12) -26	02620020
0262			(2 11 14 13) -27	02630020
0263			(2 12 12 12) -28	02640020
0264			(3 12 10 12) -29	02650020
0265			(3 15 4 15) -30	02660020
0266			(4 32) -31	02670020
0267			(5 30) -32	02680020
0268			(5 30) -33	02690020
0269			(6 28) -34	02700020
0270			(7 26) -35	02710020
0271			(9 22) -36	02720020
0272			(10 20) -37	02730020
0273			(12 16) -38	02740020
0274			(14 12) -39	02750020
0275			(17 6) -40;	02760020

DZI0415I PATTERN - INFORMATIONAL MESSAGE:
 THE PATTERN "SMALLC", AS SPECIFIED, IS 40 PELS WIDE BY 40
 PELS HIGH. THE PATTERN SIZE (A MULTIPLE OF EIGHT PELS) IS 40
 PELS WIDE BY 40 PELS HIGH.

0276	- 'ADDING GRAPHICS'	02770020
0277	POSITION ABSOLUTE 3.4 IN ABSOLUTE 2.32 IN;	02780020
0278	PLACE SEGID PRES;	02790020
0279		02800020

LINE NUM.	S O U R C E	I N P U T	S T A T E M E N T S	SEQUENCE -NUMBER-
	-----1-----	-----2-----	-----3-----4-----5-----6-----7--	
0280	POSITION ABSOLUTE	.25 IN ABSOLUTE	1.75 IN; -'PALM LOGO'	02810020
0281	PLACE SEGID	PALM;		02820020
0282				02830020
0283	POSITION ABSOLUTE	1.95 IN ABSOLUTE	2.4 IN; -INITIALS	02840020
0284	PLACE PATTERN	BIGT;		02850020
0285	POSITION LEFT	17 PELS DOWN	50 PELS;	02860020
0286	PLACE PATTERN	SMALLC;		02870020
0287	POSITION RIGHT	74 PELS DOWN	0;	02880020
0288	PLACE PATTERN	SMALLC;		02890020
0289				02900020

DZI0708I FINAL DISPOSITION:

OVERLAY FILE: NOT CREATED
SAMPLE-OVERLAY FILE: CREATED
IMAGE OPTIMIZATION: PERFORMED

MESSAGE SEVERITY SUMMARY:
0 INFORMATIONAL MESSAGES WERE SUPPRESSED.
2 INFORMATIONAL MESSAGES WERE PRINTED.
0 WARNING MESSAGES WERE SUPPRESSED.
0 WARNING MESSAGES WERE PRINTED.
0 ERROR MESSAGES WERE PRINTED.

FINAL RETURN CODE: 0

----- END OVERLAY GENERATION LANGUAGE SOURCE LISTING -----

VSE Output Listing for Overlay RCPT

- O V E R L A Y G E N E R A T I O N L A N G U A G E 3 7 0 - R 1 . 0 0 -
 ----- TIME 10:43 ---- DATE 90.200 1990-07-19 ---- PAGE 1

LINE NUM.	SOURCE INPUT STATEMENTS	SEQUENCE -NUMBER-
0001		00010018
0002	-'*****'	00020018
0003	-'* *'	00030018
0004	-'* OVERLAY GENERATION LANGUAGE / 370 (5688-191) *'	00040018
0005	-'* EXAMPLE NAME: REGISTRATION RECEIPT *'	00050018
0006	-'* (C) COPYRIGHT BY IBM 1990 *'	00060018
0007	-'* *'	00070018
0008	-'*****'	00080018
0009	-'GETTING STARTED'	00090018
0010	CONTROL NOSTORE;	00100018
0011	OVERLAY RCPT SIZE 7.25 IN 3.25 IN OFFSET .75 IN .75 IN;	00110018
0012	ORIENT 0;	00120018
0013		00130018
0014		00140018
0015	-'FONTS AND SEGMENTS'	00150018
0016	FONT FONT1 BRTR;	00160018
0017	FONT FONT2 BITR;	00170018
0018	FONT FONT3 DOTR;	00180018
0019	FONT FONT4 GT10;	00190018
0020	FONT FONT5 GT15;	00200018
0021	SEGMENT PALM PALM2;	00210018
0022	SEGMENT PRES SIGNAT;	00220018

LINES 0023 THROUGH 0288 ARE THE SAME FOR MVS, VSE AND VM

DZI0703I FINAL DISPOSITION:
 THE OVERLAY WAS NOT STORED IN THE OVERLAY LIBRARY. THE
 SAMPLE OVERLAY IS AVAILABLE FOR PRINTING. OPTIMIZATION OF
 IMAGE DATA WAS PERFORMED.

MESSAGE SEVERITY SUMMARY:
 0 INFORMATIONAL MESSAGES WERE SUPPRESSED.
 2 INFORMATIONAL MESSAGES WERE PRINTED.
 0 WARNING MESSAGES WERE SUPPRESSED.
 0 WARNING MESSAGES WERE PRINTED.
 0 ERROR MESSAGES WERE PRINTED.

FINAL RETURN CODE WAS:0

----- END OVERLAY GENERATION LANGUAGE SOURCE LISTING -----

Appendix B. Data-Set and File Allocation

MVS Data-Set Allocation

Table 3 shows the attributes of the different MVS data sets that can be used by OGL/370.

Table 3. Data-Set Attributes (MVS)

DDNAME	RECFM	LRECL	BLKSIZE	DSORG	Remarks
SYSIN	VBA	Variable	Equal to the largest record plus 4 bytes	PO or PS	If the data set has sequence numbers, see the description of sequence numbers below.
	FBA	Fixed	Multiple of LRECL		
OVRLIB	VBM	Variable 2048–32756	Equal to the largest record plus 4 bytes	PO	This data set is described in more detail below.
SYMBOLIC	FB	80 bytes	multiple of 80	PS	Refer to Appendix E, “The Symbolic Data Set and Symbolic File” on page 319.
SAMPLE	VBM	Variable 2048–32756	Equal to the largest record plus 4 bytes	PS	See below.
SYSPRINT	FBM	121	1210	PS	Standard format for SYSPRINT.

Sequence numbers

The SYSIN data set (as shown in Table 3) contains the overlay definition. If it is defined as a variable-length record data set (VBA) with sequence numbers, the numbers must appear in the first 8 bytes of each record. If it is defined as a fixed-length record data set (FBA) with sequence numbers, the numbers must appear in the last 8 bytes with each record.

Note: You do not need to inform OGL/370 of the SYSIN record format. That information is determined automatically. However, if the SYSIN data set contains sequence numbers, the EXEC statement of the Job Control Language (JCL) must indicate that fact. If the EXEC statement does not indicate the presence of sequence numbers, OGL/370 might interpret the sequence numbers as part of the overlay definition. See Appendix C, “System Dependent Information Procedures” on page 307 for details about the EXEC statement.

Allocation of OVERLIB and SAMPLE data sets

Default values (LRECL 8205 and BLKSIZE 8209) are used if no values are specified.

If LRECL and BLKSIZE are specified for only one of the data sets, the other data set automatically uses the same values.

If differing values are specified for both data sets, the lower value of each attribute is used to determine the length of the OVRLIB and SAMPLE records.

VSE File Allocation

Table 4 shows the attributes of the different VSE files that can be used by OGL/370.

Table 4. File Attributes (VSE)

FILE	RECORD ATTRIBUTES	FILE ATTRIBUTES	Remarks
SYSIPT	Fixed-length, 80-byte	Inline	If the input contains sequence numbers, see the description of sequence numbers below.
OVRLIB	As defined by the librarian	The library in which completed overlays are catalogued.	
SYMBOLIC	Fixed-length, 80-byte	Sequential disk file	

Sequence Numbers

The SYSIPT file (as shown in Table 4) contains the overlay definition and is defined as a fixed-length record file. If the SYSIPT file contains sequence numbers, the numbers must appear in the last 8 bytes of each record.

Note: You do not need to inform OGL/370 of the SYSIPT record format. That information is determined automatically. However, if the SYSIPT file contains sequence numbers, the EXEC statement of the JCL must indicate that fact. If the EXEC statement does not indicate the presence of sequence numbers, OGL/370 might interpret the sequence numbers as part of the overlay definition. See Appendix C, “System Dependent Information Procedures” on page 307 for details about the EXEC statement.

VM File Allocation

Table 5 shows the attributes of the different VM files that can be used by OGL/370.

Table 5. File Attributes (VM)

SUGGESTED FILETYPE	USE OF FILE	RECORD FORMAT	Remarks
OVERLAY	Input stream	Variable, fixed	The overlay can be created using any CMS editor.
SYMBOLIC	Symbolic input	Fixed length 80-byte	Same as above.

Sequence Numbers

The input-stream file (as shown in Table 5) contains the overlay definition. If it is defined as a variable-length record file with sequence numbers, the numbers must appear in the first 8 bytes of each record. If it is defined as a fixed-length record file with sequence numbers, the numbers must appear in the last 8 bytes of each record.

Note: You do not need to inform OGL/370 of the record format. That information is determined automatically. However, if the input file contains sequence numbers, the program invocation must indicate that fact. If the program invocation does not indicate the presence of sequence numbers, OGL/370 might interpret the sequence numbers as part of the overlay definition. See Appendix C, “System Dependent Information Procedures” on page 307 for details about the program invocation.

Appendix C. System Dependent Information Procedures

In the MVS Environment

The following JCL statements can be used to invoke OGL/370 in the MVS environment.

```
//STEP1      EXEC      PGM=DZIOVRLY,REGION=1M,  
//          PARM= '[SEQUENCE|NOSEQUence],[DEFault|ALTErnate|language code],[ADVAFP]'  
//OUTPUT1   OUTPUT   FORMDEF=F1OGL  
//SYSPRINT  DD        SYSOUT=A  
//SAMPLE    DD        SYSOUT=S,OUTPUT=*.OUTPUT1  
//OVLIB     DD        DSN=SYS1.OVLIB,DISP=OLD  
//FONTDD    DD        DSN=SYS1.FONTLIB,DISP=SHR  
//SYMBOLIC  DD        DSN=SYS1.SYMBOLIC,DISP=SHR  
//SEGDD     DD        DSN=SYS1.PSEGLIB,DISP=SHR  
//SYSIN     DD        *
```

Figure 206. Sample JCL for OGL/370 (MVS)

//STEP1

REGION=1M defines enough storage to format any of the examples in this book. Overlays containing more commands or generating more image data, may require more storage. Refer to *OGL/370: Diagnosis Guide and Reference* for guidelines on the virtual storage requirements of OGL/370.

To indicate the presence or absence of sequence numbers, choose from:

NOSEQUence The OGL/370 source file does not contain sequence numbers. All data in the file is treated as part of the command stream (default).

SEQUence The OGL/370 source file contains sequence numbers. All data in the last eight columns of fixed-length records or the first eight columns of variable-length records, is ignored.

To indicate the national language used for messages and commands, choose from:

DEFault Use the language previously defined as the “default” language.

ALTErnate Use the language previously defined as the “alternate” language.¹⁰

Language Code

Use a specific language,¹¹ choose from:

ENGLISH

GERman

JAPAnese

ADVAFP Specify **ADVAFP** to use IOCA shading for rectangular shaded areas. Otherwise IM1 shading is used.

//OUTPUT1

Identifies FORMDEF F1OGL, which is used to print the sample overlay.

10. Refer to the OGL/370 program directory to define “DEFault” and “ALTErnate” national languages.

11. English and German are used for messages and commands. If Japanese is used, only the messages are in Japanese. The commands and keywords must be entered in English.

//SYSPRINT

OGL/370 generates a source listing which contains the source statements and error messages, that result from running the program. The SYSPRINT statement identifies the data set (output spool) to be used for the source listing

//SAMPLE

Indicates that a sample overlay is to be generated and sent to SYSOUT CLASS S and printed using the FORMDEF identified in the OUTPUT1 statement.

If the sample overlay is to be saved, instead of printed, the SAMPLE statement should identify a physical sequential data set or a specific member of a PDS.

//OVLIB

Identifies the library in which the overlay is to be stored. This statement is required if **STORE** or **REPLACE** is specified in the **CONTROL** command.

Note: Concatenation is not allowed.

//FONTDD

Identifies the font library (PDS) which contains the fonts used for the overlay. You must have a DD statement for each font library that contains fonts needed for the overlay. Each of the DD statements must have a unique *DD statement name*, it is not required to use **FONTDD** as a *DD statement name*. Every DD name specified in a **FONT** command must have a corresponding DD statement in the JCL.

Note: Concatenation is not allowed.

//SYMBOLIC

Identifies the data set which contains definitions of symbolic text used in the overlay. Refer to Appendix E, "The Symbolic Data Set and Symbolic File" on page 319 for information about symbolic data sets.

//SEGDD

Identifies the segment library which contains the page segments used for the overlay. You must have a DD statement for each segment library that contains page segments needed for the overlay. Each of the DD statements must have a unique *DD statement name*, it is not required to use **SEGDD** as a *DD statement name*. Every DD name specified in a **SEGMENT** command must have a corresponding DD statement in the JCL.

Note: Concatenation is not allowed.

//SYSIN

Indicates that the OGL/370 source is inline with the JCL. This statement can also be used to identify the data set containing the OGL/370 definition statements used to build an overlay.

In the VSE Environment

The following JCS can be used to invoke OGL/370 in the VSE environment.

```
* $$ JOB      JNM=OGLVSE,CLASS=jobclass
// JOB        OGLVSE
* $$ LST      DEST=(,pdevlab),FNO=ppm,LST=X'ccu',CLASS=class
// ASSGN      SYS010,X'ccu'
// ASSGN      SYSLST,X'ccu'
// ASSGN      SYSIPT,X'ccu'
// ASSGN      SYS009,devtype,VOL=volser,SHR
// DLBL       SYMBOL,'symbolic.file.name'
// EXTENT     SYS009,volser
// LIBDEF     PHASE,SEARCH=(fontlib,seglib,pgmlib),
              CATALOG=overlib
// EXEC       PGM=DZIOVRLY,SIZE=AUTO,
              PARM='[SEQUENCE|NOSEQUENCE],[DEFAULT|ALTERNATE|language code],[ADVAFP]'
```

.....
OGL Definition Entries
.....

```
/*
/ &
* $$ E0J
```

Figure 207. Sample JCS for OGL/370 (VSE)

* \$\$ JOB

This statement designates the start of a VSE/POWER JOB. "CLASS" selects the partition in which OGL/370 is to run. The partition should have at least 850K of virtual storage available for GETVIS. Therefore, the partition to run this example should be at least 1.5M. Overlays that contain more commands or image data may require more storage than this. Refer to *OGL/370: Diagnosis Guide and Reference* for guidelines on the virtual storage requirements of OGL/370.

//JOB

Indicates the beginning of job control interaction.

* \$\$ LST

Defines the attributes of the generated output. The variable fields of these statements are:

pdevlab The label associated with the PRINTDEV macro, for initiating the printer indicated by the cuu field. See your system programmer for the PRINTDEV macro label for the printer you are using.

ppm The 4-character name of the print parameter member which contains the name of the FORMDEF to be used with the sample overlay.

Note: Under VSE, if PPFA is not available, use the FORMDEFs supplied with PSF, which contain valid FORMDEF names.

X'ccu' The physical device address of the printer you are using.

class The print class, optional.

Notes:

1. These statements and a programmer logical unit of SYS010 are required only if you are printing the sample overlay.

2. You must use the FORMDEF F1OGL, distributed with PSF, when printing the sample overlay to ensure correct positioning of the overlay.

//ASSGN

SYS010 is the programmer logical unit used only for printing the sample overlay.

//ASSGN

OGL/370 generates a listing which contains the overlay definition statements and the error messages (if any), resulting from processing the overlay. Specify the source listing with the SYSLST statement.

This statement is not required if the unit assignment is the same as the system printer default unit. It is required only if you want to print a source listing.

//ASSGN

The SYSIPT statement indicates the input file used to drive OGL/370. It contains the commands used to build an overlay.

This statement is not required if the unit assignment is the same as the system reader default unit.

//ASSGN

The ASSGN (SYS009) statement is required only if your overlay contains symbolic substitutions.

//DLBL

Describes a file created by the user which contains symbolics. Refer to Appendix E, "The Symbolic Data Set and Symbolic File" on page 319 for information on creating symbolics.

This statement is required only if your overlay contains symbolic substitutions.

//EXTENT

This statement is required only if your overlay contains symbolic substitutions.

//LIBDEF

Use the **SEARCH** parameter to identify the libraries needed for the overlay.

fontlib Identifies the font library files ID(s) which contains the fonts used in the overlay. The file IDs must be in the following form:

library.sublibrary

Note: Concatenation is not allowed.

seglib identifies the segment library file ID(s) which contains the page segments used in the overlay. The file IDs must be in the following form:

library.sublibrary

pgmlib Identifies the phase library file ID which contains the DZIOVRLY load module. If VSE/OGL resides in the default system library, you do not have to specify *pgmlib*. The file ID must be in the following form:

library.sublibrary

Use the **CATALOG** parameter to define the output library file ID in which the overlay should be stored. You can store the finished overlay in the system overlay library - PRD2.AFP, or an overlay library of your choice. The file ID must be in the following form:

library.sublibrary

Note: Use the **CATALOG** parameter only if **STORE** or **REPLACE** is specified in the **CONTROL** command.

//EXEC

This statement tells VSE to run the DZIOVRLY program.

The **SIZE** parameter indicates the amount of virtual storage to allocate for executable code. 600K is sufficient for OGL DZIOVRLY program.

The **PARM** parameter indicates whether you have specified sequence numbers in the SYSIPT input statements. Choose from:

NOSEquence The OGL/370 source file does not contain sequence numbers in columns 73 through 80. All data in the file is treated as part of the command stream, default.

SEquence The OGL/370 source file contains sequence numbers in columns 73 through 80. All characters appearing in columns 73 to 80 are ignored.

To indicate the national language used for messages and commands, choose from:

DEFault Use the language previously defined as the “default” language.¹⁰

ALternate Use the language previously defined as the “alternate” language.¹⁰

Language Code

Use a specific language,¹² choose from:

ENGLISH

GERman

JAPanese

| **ADVAFP** Specify **ADVAFP** to use IOCA shading for rectangular shaded areas. Otherwise IM1 shading is used.

12. English and German is used for messages and commands. If Japanese is used, only the messages are in Japanese. The commands and keywords must be entered in English.

In the VM Environment

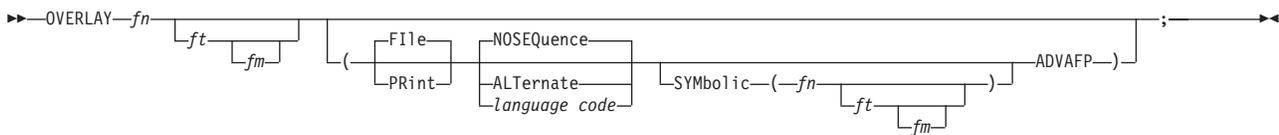
To invoke OGL/370 in the VM environment, use one of the following:

1. The OGL/370 (VM) command
2. The OGLVM EXEC.

Using the Native OGL/370 (VM) Command

The syntax of the OGL/370 VM command is:

Program Invocation Diagram for OGL/370 (VM)



fn The filename of the file containing the source input statements. This entry is required.

ft The filetype of the file containing the source input statements. This entry is optional; the default is **OVERLAY**.

fm The filemode of the file containing the source input statements. This entry is optional; the default is *****.

File/Print **File** indicates that a source listing file and a sample overlay file is created on the user's "A" disk, default.
Print indicates that a source listing file and a sample overlay file is created on the user's "A" disk, and that the sample overlay file is printed.
This entry is optional.

NOSEquence/SEquence

NOSEquence The OGL/370 source file does not contain sequence numbers, default.

SEquence The OGL/370 source file contains sequence numbers. All data is ignored.

To indicate the national language used for messages and commands, choose from:

Default Use the language previously defined as the "default" language.¹⁰

ALternate Use the language previously defined as the "alternate" language.¹⁰

Language Code

Use a specific language,¹³ choose from:

ENGLISH
GERman
JAPANEse

Notes:

1. Before you issue the **OVERLAY** command for the first time during a session, you must issue the following command:

```
GLobal TXTLIB OGLENG OGLGER OGLJAP
```

13. English and German is used for messages and commands. If Japanese is used, only the messages are in Japanese. The commands and keywords must be entered in English.

A convenient way of automatically doing this is to add the command to your PROFILE EXEC.

2. To see Japanese messages (which use DBCS), your terminal must be in FULLSCREEN mode. You can set this on by issuing the following CMS command:

```
SET FULLSCREEN ON
```

Symbolic File

If a symbolic file is to be used, enter SYMBolic and as much of the following symbolic file ID in parentheses as necessary. For additional information about creating a symbolic file for use with your overlay, see Appendix E, "The Symbolic Data Set and Symbolic File" on page 319.

<i>fn</i>	The filename of the file containing the symbolic entries. This entry is required if the SYMBOLIC keyword is used.
<i>ft</i>	The filetype of the file containing the symbolic entries. This entry is optional; the default is SYMBOLIC .
<i>fm</i>	The filemode of the file containing the symbolic entries. This entry is optional; the default is *.
ADVAFP	Specify ADVAFP to use IOCA shading for rectangular shaded areas. Otherwise IM1 shading is used.

Using OGLVM EXEC

When you enter OGLVM on the VM command line, the panel illustrated in Figure 208 is displayed. This panel requires the same information as the VM command, but is easier to understand and use.

Fill in the appropriate fields and press ENTER to run the command.

```
***** Overlay Generation Language / 370 *****
Use the tab keys to go to the next field.
Source file: Press the enter key when finished.
  Filename    ==> (filename is required)
  Filetype    ==> OVERLAY (optional, default is "OVERLAY")
  Filemode    ==> * (optional, default is "*")
Options:
  Output      ==> F (f)=file or (p)=print
  Sequence numbers ==> N (y)=yes or (n)=no
  Language    ==> DEF (def)=default, (alt)=alternate
              (or specify language code)
Symbolic file: ===== OPTIONAL =====
  Filename    ==> (if used, filename is required)
  Filetype    ==> (optional, default is "SYMBOLIC")
  Filemode    ==> (optional, default is "*")
PF1=HELP PF3=END PF12=CURSOR
```

Figure 208. Submitting an Overlay for Compilation

Using a PSF EXEC

If you do not have PSF/VM, you may write an exec with the name PSF EXEC to set up any necessary job control and to ship the sample overlay file to another system for printing. If a PSF EXEC is accessed, OGL/370 invokes it.

The invocation of the exec is in the form of a CMS command:

EXEC PSF *fn ft fm* (CC

where *fn ft fm* is the file ID of the sample overlay file.

OGI/370 calls a PSF EXEC, but does not check any return codes from the exec. It is left up to the exec writer to include any desired error handling. If a PSF EXEC is called, OGI/370 reports the sample overlay as having been printed.

Appendix D. Merging Overlays and Variable Data

In many printing applications, the OGL/370 licensed program is used to build an overlay that is then merged with formatted variable data (a “page” of data). Overlay PREREG, for example, could be merged with data for each student (for instance name, address, enrollment). *Page Printer Formatting Aid User’s Guide and Reference* describes how to merge overlays with the corresponding variable, even if they are rotated on the page.¹⁴

PMF and PPFA are IBM licensed programs that code the information needed to merge variable data with an overlay. This information is contained in two sets of instructions called a PAGEDEF and a FORMDEF. A PAGEDEF formats variable data and positions it relative to the page origin (analogous to the overlay origin). A FORMDEF positions the page origin on the paper (analogous to the **OFFSET** subcommand in the **OVERLAY** command).

Coordinating Form Definitions and Page Definitions

For variable data to appear correctly on the medium overlay, the overlay definition, FORMDEF, and PAGEDEF must be coordinated. Coordination involves making sure that:

- The page size (in the PAGEDEF) and the overlay size (in the **OVERLAY** command) are the same.
- The page position (in the FORMDEF) and the overlay origin (in the **OVERLAY** command) are the same.
- The page direction (in the PAGEDEF) and the overlay orientation (**ORIENT** command) are the same.
- Corresponding variable text and overlay text are consistently specified.

The first three items (size, offset, and orientation) need no further explanation. The remainder of this appendix deals with the last item, text.

Text Specification: Because of differences in tasks, OGL/370 specifies text somewhat differently than do PMF and or PPFA. Figure 209 on page 316 shows the possible combinations of format and orientation. The overlay text (**Last First MI**) and its OGL/370 description of orientation and format are shown in **boldface**. The variable text (Smith John D) and its PMF and PPFA description of direction and rotation are shown in medium face.

14. Keep in mind that to print text in a given format you must have the font in the correct orientation (see Appendix F, “Matching Fonts with Text Formatting” on page 323).

Sample

Smith	John	D
Last	First	MI

Overlay Generation
Language

Print Management
Facility and PPFA

0 MODERN

ACROSS 0

Smith	John	D
Last	First	MI

90 MODERN

DOWN 0

MI	First	Last
D	John	Smith

180 MODERN

BACK 0

D	John	Smith
MI	First	Last

270 MODERN

UP 0

Figure 209. Variable and Overlay Text (Part 1 of 2)

Sample	Overlay Generation Language	Print Management Facility and PPFA																														
<table border="1"> <tr><td>F</td><td>S</td></tr> <tr><td>i</td><td>m</td></tr> <tr><td>r</td><td>i</td></tr> <tr><td>s</td><td>t</td></tr> <tr><td>t</td><td>h</td></tr> <tr><td colspan="2"> </td></tr> <tr><td>L</td><td>J</td></tr> <tr><td>a</td><td>o</td></tr> <tr><td>s</td><td>h</td></tr> <tr><td>t</td><td>n</td></tr> <tr><td colspan="2"> </td></tr> <tr><td>M</td><td>D</td></tr> <tr><td>I</td><td></td></tr> </table>	F	S	i	m	r	i	s	t	t	h			L	J	a	o	s	h	t	n			M	D	I		0 COLUMN or 0 TATE	DOWN 270				
F	S																															
i	m																															
r	i																															
s	t																															
t	h																															
L	J																															
a	o																															
s	h																															
t	n																															
M	D																															
I																																
<table border="1"> <tr><td>I</td><td>t</td><td>s</td><td>r</td><td>e</td><td>s</td></tr> <tr><td>M</td><td>h</td><td>o</td><td>n</td><td>J</td><td>S</td></tr> <tr><td>D</td><td>n</td><td>h</td><td>o</td><td>h</td><td>m</td></tr> </table>	I	t	s	r	e	s	M	h	o	n	J	S	D	n	h	o	h	m	90 COLUMN or 90 TATE	BACK 270												
I	t	s	r	e	s																											
M	h	o	n	J	S																											
D	n	h	o	h	m																											
<table border="1"> <tr><td>D</td><td>I</td></tr> <tr><td>u</td><td>t</td></tr> <tr><td>y</td><td>s</td></tr> <tr><td>o</td><td>a</td></tr> <tr><td>J</td><td>L</td></tr> <tr><td colspan="2"> </td></tr> <tr><td>h</td><td>t</td></tr> <tr><td>S</td><td>r</td></tr> <tr><td>m</td><td>!</td></tr> <tr><td>S</td><td>F</td></tr> </table>	D	I	u	t	y	s	o	a	J	L			h	t	S	r	m	!	S	F	180 COLUMN or 180 TATE	UP 270										
D	I																															
u	t																															
y	s																															
o	a																															
J	L																															
h	t																															
S	r																															
m	!																															
S	F																															
<table border="1"> <tr><td>S</td><td>J</td><td>D</td></tr> <tr><td>m</td><td>o</td><td></td></tr> <tr><td>i</td><td>h</td><td>n</td></tr> <tr><td>t</td><td>n</td><td></td></tr> <tr><td>h</td><td></td><td></td></tr> <tr><td colspan="3"> </td></tr> <tr><td>F</td><td>L</td><td>M</td></tr> <tr><td>i</td><td>a</td><td>I</td></tr> <tr><td>r</td><td>s</td><td></td></tr> <tr><td>s</td><td>t</td><td></td></tr> </table>	S	J	D	m	o		i	h	n	t	n		h						F	L	M	i	a	I	r	s		s	t		270 COLUMN or 270 TATE	ACROSS 270
S	J	D																														
m	o																															
i	h	n																														
t	n																															
h																																
F	L	M																														
i	a	I																														
r	s																															
s	t																															

Figure 209. Variable and Overlay Text (Part 2 of 2)

Translating Measurements

OGL/370 translates all measurements into a whole number of pels. However, some measurements that you can specify in an overlay definition result in a fractional number of pels (for example, 0.17 inches equals 40.8 pels). If a measurement is a fractional number of pels, OGL/370 truncates the measurement. Thus, if you specify:

```
POSITION ABSOLUTE @ ABSOLUTE .17 IN;
```

the new position is 40 pels from the top of the overlay. However, if a **REPEAT** subcommand specifies a rule spacing of 0.17 inches, OGL/370 truncates each spacing and carries the difference to the next spacing. For example, if you position at 40 pels from the top margin of the overlay and specify a rule repeated 4 times and **SPACED .17 IN**, this is the result:

Rule	Distance from Top Margin (Pels)	Distance from Previous Line (Pels)
1st	40	—
2nd	80	40
3rd	121	41
4th	162	41
5th	203	41
6th	244	41
7th	284	40

PMF treats fractions of pels somewhat differently. It truncates the value, but does not save the difference. Be careful, or the overlay and the variable data will not align correctly. If, in a **PAGEDEF** definition, you specify 7 lines of text to be spaced 0.17 inches starting 0.17 inches below the top margin, this is the result:

Line	Distance from Top Margin (Pels)	Distance from Previous Line (Pels)
1st	40	—
2nd	80	40
3rd	120	40
4th	160	40
5th	200	40
6th	240	40
7th	280	40

As you can see, the cumulative effects of the difference could be significant. You can easily avoid such misalignment. When PMF and OGL/370 must be coordinated, use a unit of measurement that does not result in a fractional number of pels.

Appendix E. The Symbolic Data Set and Symbolic File

The symbolic data set (in MVS) and the symbolic file (in VSE and VM) contain text substitutions for symbols that appear in the text strings of the **DRAWBOX**, **DRAWCIRCLE**, and **SETTEXT** commands.

It contains a one-for-one correlation of symbolics to replacement text. The symbolic function allows you to reference symbolically previously defined code points for characters that are not normally found on terminal keyboards. You can also use this function to substitute a common text string (for example, a copyright statement) for a predefined symbolic record.

For example, if OGL/370 finds the symbolic "&ABOUT" in the text string of a **SETTEXT** command, the program searches the symbolic data set or file for the "ABOUT" symbolic. During this search, if OGL/370 finds a record entry, "&ABOUT X'9E," the program substitutes the hexadecimal code point value of 9E for the symbolic "&ABOUT" in the text string.

You can also use non-numeric characters in the symbolic data set or file. For example, if in the symbolic data set or file you set the symbolic "IUO C'Internal Use Only", any time OGL/370 finds the symbolic "&IUO" in a text string, the program substitutes the words "Internal Use Only".

The symbolic data set must contain 80-byte records that adhere to the following rules and format:

- Each record in the symbolic data set must contain both the symbolic search argument and the replacement text; together they cannot exceed 80 bytes.
- You must use one or more spaces to separate the symbolic search argument from the replacement text.
- The symbolic name must not have blanks between the "&" and the ending period.
- The symbolic search argument must start in column 1 and can be from 1 to n characters in length (where n is any number, as long as the record does not exceed 80 bytes).
- The replacement text can be from 1 to n bytes long (where n is any number as long as the record does not exceed 80 bytes).
- You can represent text in either character or hexadecimal form using the following format:
 - X'hexadecimal data'
 - C'alphanumeric data'
- The text type must be followed immediately by an apostrophe (').

Note: The closing apostrophe ('), or the end of the 80-byte record, terminates the symbolic replacement data.

Using the Symbolic Data Set in MVS

Things to know when using the symbolic data set in MVS:

- You must define the symbolic data set with the //SYMBOLIC DD statement of the JCL necessary to submit an overlay. Refer to Appendix C, "System Dependent Information Procedures" on page 307.
- The symbolic data set must be a sequential data set with a record format of fixed-length, 80-byte logical records.
- OGL/370 uses the Queued Sequential Access Method (QSAM) to read the symbolic data set.

Using the Symbolic File in VSE

Things to know when using the symbolic file in VSE:

- You must define the symbolic file with the //DLBL SYMBOL statement of the JCS necessary to submit an overlay in VSE. Refer to Appendix C, "System Dependent Information Procedures" on page 307.
- The symbolic file must be a sequential file with a record format of fixed-length, 80-byte physical records.

- You can only use one symbolic file per job.

Using the Symbolic File in VM

Things to know when using the symbolic file in VM:

- Define the symbolic file in the program invocation in VM. Refer to Appendix C, “System Dependent Information Procedures” on page 307.
- The symbolic file must be a sequential file with a record format of fixed-length, 80-byte physical records.
- You can only use one symbolic file per job.

Symbolic Data Sets and Symbolic Files Containing Double-Byte Characters

Three types of entries can be made in symbolic data sets (in MVS) or symbolic files (in VSE and VM).

Type 1 This type of entry can be made from any keyboard device. It consists of three parts: the symbolic name, the text type (X), and the hexadecimal code between apostrophes. In the preceding examples, the text string '&NAME1.&NAME2.' corresponds to these entries in a symbolic data set:

NAME1 X'4546'

NAME2 X'454B'

Type 2 This type of entry can be made from a Kanji keyboard device. It consists of two parts: the symbolic name and the Kanji characters between apostrophes.

Note: Use a “#” to indicate that the entire symbolic record is entered as double-byte characters.

The symbolic name is entered as a string of two-byte characters, with the first byte of each character being a hexadecimal X'42'. The apostrophes are also preceded by X'42'. In the preceding examples, the text string " correspond to the following entries in a symbolic data set:

³N³A³M³E³1³ 六³

³N³A³M³E³2³ 百³

³ Indicates a hexadecimal '42'

Type 3 This type of entry is made from an IBM 5550 or PS/55 Work Station. It consists of three parts: the symbolic name, the text type (G), and the Kanji characters between apostrophes. In the preceding examples, the text string '&NAME1.&NAME2.' corresponds to the following entries in a symbolic data set or symbolic file:

NAME1 G'<六>'

NAME2 G'<𐄆>'

< Indicates a shift-out character

> Indicates a shift-in character

Latin text (and therefore symbolic names) is not allowed between the apostrophes. Do not include blanks between apostrophes.

Appendix F. Matching Fonts with Text Formatting

Note: The information in this section applies only to the IBM 3800 Printing Subsystem Model 3 and Model 8.

If your font library has font X1GT10, you could define a 0° overlay orientation and 0° **MODERN** text using the font *member ID* GT10; assuming that the rest of your definition is correct, the text is printed as specified. However, if you ask for 0° **COLUMN** text, it might not be printed. For that text to print, your font library must contain font GT10 in the correct orientation: it must contain font XEGT10.

Figure 210 on page 324 relates the font prefixes to the orientation and format combinations *relative to the paper*. The italicized phrase is important. For the IBM 3800 Printing Subsystem Model 3 and Model 8, if you specified text as 0° **MODERN** but rotated the overlay to 90°, the text would be 90° **MODERN** relative to the paper. Font X2GT10 would then be required. For IBM AFP printers, you only need to be concerned that you have the font you want, because the font prefix (“X0”) is the same for all IBM AFP printer fonts.

Sample	Orientation & Format	Single-byte Font Prefix		Double-byte Font Prefix	
		3800	3820	3800	3820
ABCD	0° MODERN	X1	X0	X1	X0
ABCD	90° MODERN	X2	X0	NONE	X0
ABCD	180° MODERN	X3*	X0	NONE	X0
ABCD	270° MODERN	X4	X0	NONE	X0
A B C D	0° COLUMN or TATE	XE	X0	X1	X0
A B C D	90° COLUMN or TATE	XF*	X0	NONE	X0
D C B A	180° COLUMN or TATE	XG	X0	NONE	X0
A B C D	270° COLUMN or TATE	XD	X0	NONE	X0

* These prefixes are not supported on the 3800 Model 3.

Figure 210. Font Prefixes for Orientation and Format Combinations

Appendix G. Shade Patterns and Types

UGL/370 provides 32 shades, in two patterns; **STANDARD** (default) or **SCREEN**. Shades can be specified in **DRAWBOX**, **DRAWCIRCLE**, **DRAWPATH**, and **PLACE PATTERN**. After specifying the shade pattern, shade type is selected either by entering one of the five named types or by entering a percent number, shown in the Figure 211 on page 326 and Figure 212 on page 330.

Since the results may vary on different printers, the source code used to create the samples is provided so that you may print them on your printer.



Figure 211. Shade Pattern — **STANDARD**

The VM source code for **STANDARD** shading patterns:

```
-----
-' Samples of OGL/370 STANDARD shading pattern
-----

OVERLAY shstan
  SIZE 6.5 in 10 in
  OFFSET 0 in 0;

CONTROL replace;

FONT f1
  a0559c
  FILETYPE font3820;

FONT label
  a0558c
  FILETYPE font3820;

SETUNITS 1 in;

POSITION .1 in .1 in;

-----
-' Box labels - first column
-----
DRAWBOX 1 .4 0
  repeat down 15 spaced .1
  withtext box 1 line label '0-5%'
    line label 'XLIGHT'
  withtext box 2 line label '6-8%'
  withtext box 3 line label '9-11%'
  withtext box 4 line label '12-14%'
  withtext box 5 line label '15-17%'
  withtext box 6 line label '18-20%'
  withtext box 7 line label '21-23%'
  withtext box 8 line label '24-26%'
    line label 'LIGHT'
  withtext box 9 line label '27-29%'
  withtext box 10 line label '30-32%'
  withtext box 11 line label '33-35%'
  withtext box 12 line label '36-38%'
  withtext box 13 line label '39-41%'
  withtext box 14 line label '42-44%'
  withtext box 15 line label '45-47%'
  withtext box 16 line label '48-50%'
    line label 'MEDIUM';

-----
-' Boxes with shading and text - first column
-----
POSITION right 1 right 0;
DRAWBOX 1.5 .4 0
  repeat down 15 spaced .1
  shade box 1 standard 3
    box 2 standard 6
    box 3 standard 9
    box 4 standard 12
    box 5 standard 15
    box 6 standard 18
    box 7 standard 21
    box 8 standard 24
    box 9 standard 27
    box 10 standard 30
    box 11 standard 33
    box 12 standard 36
    box 13 standard 39
```

```

        box 14 standard 42
        box 15 standard 45
        box 16 standard 48
withtext box 1 line f1 '1234567890'
withtext box 2 line f1 '1234567890'
withtext box 3 line f1 '1234567890'
withtext box 4 line f1 '1234567890'
withtext box 5 line f1 '1234567890'
withtext box 6 line f1 '1234567890'
withtext box 7 line f1 '1234567890'
withtext box 8 line f1 '1234567890'
withtext box 9 line f1 '1234567890'
withtext box 10 line f1 '1234567890'
withtext box 11 line f1 '1234567890'
withtext box 12 line f1 '1234567890'
withtext box 13 line f1 '1234567890'
withtext box 14 line f1 '1234567890'
withtext box 15 line f1 '1234567890'
withtext box 16 line f1 '1234567890';

```

```

-----'
-' Box labels - second column
-----'

```

```

POSITION right 2 in right 0;
DRAWBOX 1 .4 0
  repeat down 15 spaced .1
  withtext box 1 line label '51-53%'
  withtext box 2 line label '54-56%'
  withtext box 3 line label '57-59%'
  withtext box 4 line label '60-62%'
  withtext box 5 line label '63-65%'
  withtext box 6 line label '66-68%'
  withtext box 7 line label '69-71%'
  withtext box 8 line label '72-74%'
    line label 'DARK'
  withtext box 9 line label '75-77%'
  withtext box 10 line label '78-80%'
  withtext box 11 line label '81-83%'
  withtext box 12 line label '84-86%'
  withtext box 13 line label '87-89%'
  withtext box 14 line label '90-92%'
  withtext box 15 line label '93-95%'
  withtext box 16 line label '96-100%'
    line label 'XDARK';

```

```

-----'
-' Boxes with shading and text - second column
-----'

```

```

POSITION right 1 right 0;
DRAWBOX 1.5 .4 0
  repeat down 15 spaced .1
  shade box 1 standard 51
    box 2 standard 54
    box 3 standard 57
    box 4 standard 60
    box 5 standard 63
    box 6 standard 66
    box 7 standard 69
    box 8 standard 72
    box 9 standard 75
    box 10 standard 78
    box 11 standard 81
    box 12 standard 84
    box 13 standard 87
    box 14 standard 90
    box 15 standard 93
    box 16 standard 96
  withtext box 1 line f1 '1234567890'

```

```
withtext box 2 line f1 '1234567890'  
withtext box 3 line f1 '1234567890'  
withtext box 4 line f1 '1234567890'  
withtext box 5 line f1 '1234567890'  
withtext box 6 line f1 '1234567890'  
withtext box 7 line f1 '1234567890'  
withtext box 8 line f1 '1234567890'  
withtext box 9 line f1 '1234567890'  
withtext box 10 line f1 '1234567890'  
withtext box 11 line f1 '1234567890'  
withtext box 12 line f1 '1234567890'  
withtext box 13 line f1 '1234567890'  
withtext box 14 line f1 '1234567890'  
withtext box 15 line f1 '1234567890'  
withtext box 16 line f1 '1234567890';
```



Figure 212. Shade Pattern — **SCREEN**

The VM source code for **SCREEN** shading patterns:

```
-----  
-' Samples of OGL/370 SCREEN shading pattern  
-----  
  
OVERLAY shscrn  
  SIZE 6.5 in 10 in  
  OFFSET 0 in 0;  
  
CONTROL replace;  
  
FONT f1  
  a0559c  
  FILETYPE font3820;  
  
FONT label  
  a0558c  
  FILETYPE font3820;  
  
SETUNITS 1 in;  
  
POSITION .1 in .1 in;  
  
-----  
-' Box labels - first column  
-----  
DRAWBOX 1 .4 0  
  repeat down 15 spaced .1  
  withtext box 1 line label '0-5%'  
    line label 'XLIGHT'  
  withtext box 2 line label '6-8%'  
  withtext box 3 line label '9-11%'  
  withtext box 4 line label '12-14%'  
  withtext box 5 line label '15-17%'  
  withtext box 6 line label '18-20%'  
  withtext box 7 line label '21-23%'  
  withtext box 8 line label '24-26%'  
    line label 'LIGHT'  
  withtext box 9 line label '27-29%'  
  withtext box 10 line label '30-32%'  
  withtext box 11 line label '33-35%'  
  withtext box 12 line label '36-38%'  
  withtext box 13 line label '39-41%'  
  withtext box 14 line label '42-44%'  
  withtext box 15 line label '45-47%'  
  withtext box 16 line label '48-50%'  
    line label 'MEDIUM';  
  
-----  
-' Boxes with shading and text - first column  
-----  
POSITION right 1 right 0;  
DRAWBOX 1.5 .4 0  
  repeat down 15 spaced .1  
  shade box 1 screen 3  
    box 2 screen 6  
    box 3 screen 9  
    box 4 screen 12  
    box 5 screen 15  
    box 6 screen 18  
    box 7 screen 21  
    box 8 screen 24  
    box 9 screen 27  
    box 10 screen 30  
    box 11 screen 33  
    box 12 screen 36  
    box 13 screen 39
```

```

        box 14 screen 42
        box 15 screen 45
        box 16 screen 48
withtext box 1 line f1 '1234567890'
withtext box 2 line f1 '1234567890'
withtext box 3 line f1 '1234567890'
withtext box 4 line f1 '1234567890'
withtext box 5 line f1 '1234567890'
withtext box 6 line f1 '1234567890'
withtext box 7 line f1 '1234567890'
withtext box 8 line f1 '1234567890'
withtext box 9 line f1 '1234567890'
withtext box 10 line f1 '1234567890'
withtext box 11 line f1 '1234567890'
withtext box 12 line f1 '1234567890'
withtext box 13 line f1 '1234567890'
withtext box 14 line f1 '1234567890'
withtext box 15 line f1 '1234567890'
withtext box 16 line f1 '1234567890';

```

```

-----'
-' Box labels - second column
-----'

```

```

POSITION right 2 in right 0;
DRAWBOX 1 .4 0
  repeat down 15 spaced .1
  withtext box 1 line label '51-53%'
  withtext box 2 line label '54-56%'
  withtext box 3 line label '57-59%'
  withtext box 4 line label '60-62%'
  withtext box 5 line label '63-65%'
  withtext box 6 line label '66-68%'
  withtext box 7 line label '69-71%'
  withtext box 8 line label '72-74%'
    line label 'DARK'
  withtext box 9 line label '75-77%'
  withtext box 10 line label '78-80%'
  withtext box 11 line label '81-83%'
  withtext box 12 line label '84-86%'
  withtext box 13 line label '87-89%'
  withtext box 14 line label '90-92%'
  withtext box 15 line label '93-95%'
  withtext box 16 line label '96-100%'
    line label 'XDARK';

```

```

-----'
-' Boxes with shading and text - second column
-----'

```

```

POSITION right 1 right 0;
DRAWBOX 1.5 .4 0
  repeat down 15 spaced .1
  shade box 1 screen 51
    box 2 screen 54
    box 3 screen 57
    box 4 screen 60
    box 5 screen 63
    box 6 screen 66
    box 7 screen 69
    box 8 screen 72
    box 9 screen 75
    box 10 screen 78
    box 11 screen 81
    box 12 screen 84
    box 13 screen 87
    box 14 screen 90
    box 15 screen 93
    box 16 screen 96
  withtext box 1 line f1 '1234567890'

```

```
withtext box 2 line f1 '1234567890'  
withtext box 3 line f1 '1234567890'  
withtext box 4 line f1 '1234567890'  
withtext box 5 line f1 '1234567890'  
withtext box 6 line f1 '1234567890'  
withtext box 7 line f1 '1234567890'  
withtext box 8 line f1 '1234567890'  
withtext box 9 line f1 '1234567890'  
withtext box 10 line f1 '1234567890'  
withtext box 11 line f1 '1234567890'  
withtext box 12 line f1 '1234567890'  
withtext box 13 line f1 '1234567890'  
withtext box 14 line f1 '1234567890'  
withtext box 15 line f1 '1234567890'  
withtext box 16 line f1 '1234567890';
```

Appendix H. Printer Characteristics

OGL/370 is designed for use on several IBM printers. Some of these printers may not be able to print the overlay exactly as it was specified in the overlay definition. For example, shading levels may differ or some printers may not support specific text orientations.

This appendix describes some of the printer characteristics that may affect the way you design your overlays. For more information about printer characteristics, refer to the *Advanced Function Printing: Printer Information*, or appropriate documentation for the printer that interests you.

Printable Area

In OGL/370, the printable area is the area on the page where an overlay can be positioned. Different printers have different printable areas. The printable area may also depend on the paper size and orientation you use.

The **OVERLAY** command (described in “Beginning the Overlay (**OVERLAY**)” on page 27) allows you to specify the offset and dimensions of the overlay so that it will fit within the printable area of your printer.

The printable areas for various printers are described in *Advanced Function Printing: Printer Information*.

Text Direction Restrictions

OGL/370 provides the capability to orient the overlay relative to the sheet of paper and to orient text relative to the overlay. In addition, text may be formatted on the page as **MODERN**, **COLUMN**, or **TATE**. Not all printers can print all combinations of overlay orientation, text orientation, and text format. For instance, the 3800 Printer cannot print text that goes from right to left on the paper, such as **MODERN** 180° text in an overlay that is oriented at 0° with respect to the paper.

Printer Storage Limitations

The data needed to print an overlay is loaded into printer storage. With unusually complex overlays, the storage requirements of the overlay may approach or exceed the storage limitations of the printer. If the storage limitations are exceeded, the overlay will not print. If the storage used is close to the limitations, printing may be slowed somewhat. In such cases, you should consider analyzing the storage requirements for the overlay and redesigning it. You should also consider purchasing more raster image storage for your printer if this is appropriate.

Appendix J, “Storage Summary” on page 341 explains storage limitations in more detail and directs you to the information that you need to calculate storage requirements. The *Advanced Function Printing: Diagnosis Guide* contains an explanation for the procedure for calculating storage requirements. “Specifying Storage, Message, and **SOSI** Options (**CONTROL**)” on page 26 explains how to request the information needed to analyze storage requirements.

Appendix I. OGL/370 Keywords

A keyword is a word in OGL/370 that must be entered exactly as shown. Keywords cannot be used as names in a font, segment, definition, or overlay.

The following is a list of keywords:

ABS	CYAN	FONT	NOSOSI	SIXSTAR
ABSOLUTE	C1VAL	GIF	NOSTORE	SIZE
ACROSS	C2VAL	GOCA	NOSUMM	SMALL
ALL	DARK	GREEN	NOSUMMARY	SOLID
AUTO	DASHDOT	GROUP	NOTRACE	SOSI
AXIS	DASHED	GVAL	NOUND	SOSI1
AXES	DDNAME	HALF	NOUNDER	SOSI2
BALANCE	DEF	HCOLOR	NOUNDERLINE	SPACED
BCOCA	DEFAULT	HEIGHT	OBJECT	SQUARE
BCOLOR	DEFINE	HEX	OBNAME	STANDARD
BDRCOLOR	DEPTH	HIGHLIGHT	OBTYPE	STORE
BL	DIAGONAL	HORIZONTAL	OCA	SUMM
BLACK	DIAMETER	HORZLN	OCD	SUMMARY
BLTR1	DIAMOND	IN	OEG	TATE
BLTR2	DOT	INCH	OFFSET	TEXTM
BLUE	DOT01	INCHES	ORIENT	TEXTMARGIN
BMP-OS2	DOT02	IOCA	OTHER	TIFF
BMP-WIN	DOT03	JFIF	OVERLAY	TL
BOLD	DOT04	JUSTIFY	PATH	TLBR1
BORDERWEIGHT	DOT05	KVAL	PATHEND	TLBR2
BOTH	DOT06	LARGE	PATTERN	TO
BOTTOM	DOT07	LASTNO	PCL	TOP
BOTTOMLEFT	DOT08	LEFT	PCX	TOPLEFT
BOTTOMRIGHT	DOTTED	LIGHT	PEL	TOPRIGHT
BOX	DOUBLEDOT	LINE	PELS	TR
BR	DOWN	LINESP	PLACE	TRACE
BROWN	DRAWB	LOC	PLUS	TRACEALL
BVAL	DRAWBOX	LOCATION	POINTS	TRIM
BW	DRAWC	LONGDASH	POS	TRIMMED
CBDUMP	DRAWCIRCLE	LPI	POSING	TYPENAME
CENTER	DRAWG	LVAL	POSITION	UCOLOR
CHAR	DRAWGRAPHIC	MAGENTA	POSITIONING	UND
CHARSET	DRAWM	MARKER	PSEG	UNDER
CIELAB	DRAWMASK	MAX	QUARTER	UNDERLINE
CIRCLE	DRAWP	MEDIUM	RED	UP
CL	DRAWPATH	MILLIMETER	REPEAT	VCOLOR

CLOSE	DRAWR	MILLIMETERS	REPLACE	VERTICAL
CMYK	DRAWRULE	MIRROR	RGB	VERTLN
I CODEPAGE	DSHDBLDOT	MITER	RIGHT	WARN
COL	EIGHTSTAR	MM	ROUNDED	WHOLE
COLOR	ELLIPSE	MOD	RVAL	WIDTH
COLUMN	ENCODED	MODCA	SCALE	WITHT
CONNECTION	END	MODERN	SCREEN	WITHTEXT
CONTROL	ENDDEF	MVAL	SEGID	XDARK
CORNERLENGTH	EPS	NEG	SEGMENT	XLIGHT
COVERAGE	ERROR	NEGATIVE	SETT	YELLOW
CP	FILETYPE	NOFILL	SETTEXT	YVAL
CPI	FILL	NOMIRROR	SETU	
CROSS	FILLDMND	NON	SETUNITS	
I CS	FILLETS	NONEG	SHADE	
CVAL	FILLSQR	NONNEGATIVE	SHORTDASH	

Modifying the Command Keywords

OGL/370 lets you specify synonyms for keywords. You might want to do this in two situations:

- You might want to use an entirely different keyword in your overlay definitions. For example, if you specify **BOXTEXT** as a synonym for **WITHTEXT**, you could use either of these words in a **DRAWBOX** command to add text to the box.
- You might want to add abbreviations for a keyword. For example, if you specify **DBOX** as a synonym for **DRAWBOX**, any of the following keywords could be used for the same command: **DRAWBOX**, **DRAWB**, **DBOX**.

Note: A keyword that already exists in OGL/370 cannot be defined as a synonym. For example, you could not define **BOX** as a synonym for **DRAWBOX**, because **BOX** is already a keyword (to specify the box to which shade or text is to be added).

To add synonyms for the keywords, you must edit module DZILxxxK, the keyword CSECT for the language you are using. The language is indicated by the value of 'xxx' (**ENG** for English, **GER** for German, and **JAP** for Japanese), and this is an assembler module shipped with OGL/370. The :q.keyword code:eq. section of DZILxxxK associates one or more keywords (KWWORD) with keyword codes (KEY CODE). The following example shows the subsection that defines keyword codes 21 to 25, "Functional Commands". For each keyword, two parameters (CODE= and KWWORD=) are passed to the KEY macro.

```

* *****
* KEYWORD CODES 21 - 30: FUNCTIONAL COMMANDS
* *****
    KEY CODE=21,KWORD=DRAWMASK
    KEY CODE=21,KWORD=DRAWM
    KEY CODE=22,KWORD=DRAWRULE
    KEY CODE=22,KWORD=DRAWR
    KEY CODE=23,KWORD=DRAWBOX
    KEY CODE=23,KWORD=DRAWB
    KEY CODE=24,KWORD=POSITION
    KEY CODE=24,KWORD=POS
    KEY CODE=25,KWORD=SETTEXT
    KEY CODE=25,KWORD=SETT
* *****

```

Notice that **DRAWBOX** and **DRAWB** are both associated with the same keyword code, 23. To specify **DDBOX** as another abbreviation for **DRAWBOX**, you must associate it with keyword code 23. Simply add the following macro invocation:

```
KEY CODE=23 KWORD=DBOX
```

The list on page Table 6 shows all the code numbers and the associated keywords. Note the following restrictions when making additions to the CSECT:

- Do not change, add, or delete any code numbers.
- All synonyms for a particular keyword code must be grouped together.
- All codes must be in ascending order in the CSECT.
- The maximum length for a keyword is 40 characters.
- The first character of the keyword cannot be a hyphen (-).
- The keyword must not include an apostrophe ('), a left parenthesis ((), or a semi-colon (;).

When you have completed the additions to the CSECT, DZILxxxK must be reassembled and link edited. For instructions on link editing, see *OS/VS VM/370 Assembler Programmer's Guide* and *VSE/Advanced Functions System Control Statements*.

Keywords and Code Numbers

Table 6. Keywords and Code Numbers

Code	Keyword	Code	Keyword	Code	Keyword	Code	Keyword
01	CONTROL	92	UP	169	JUSTIFY	238	C2VAL
02	OVERLAY	93	DOWN	170	LARGE	239	HCOLOR
03	ORIENT	100	GROUP	171	LASTNO	240	VCOLOR
04	FONT	101	PATTERN	173	MAX	241	UCOLOR
05	SEGMENT	102	ENCODED	174	MITER	242	BCOLOR
10	CBDUMP	103	SEGID	176	PATHEND	242	BDRCOLOR
16	SETUNITS	104	MIRROR	178	QUARTER	245	OBJECT
16	SETU	105	NOMIRROR	180	ROUNDED	246	OBNAME
17	DRAWCIRCLE	106	NEGATIVE	181	SMALL	247	OBTYPE
17	DRAWC	106	NEG	182	SOSI1	248	IOCA
18	DRAWPATH	107	NONNEGATIVE	183	SOSI2	249	GOCA
18	DRAWP	107	NONEG	184	SQUARE	250	BCOCA
21	DRAWMASK	110	WITHTEXT	185	TRIM	251	PSEG
21	DRAWM	110	WITHT	185	TRIMMED	252	OTHER

Table 6. Keywords and Code Numbers (continued)

Code	Keyword	Code	Keyword	Code	Keyword	Code	Keyword
22	DRAWRULE	111	BOX	186	WHOLE	253	MODCA
22	DRAWR	112	REPEAT	187	BOTTOMLEFT	254	OEG
23	DRAWBOX	113	SHADE	187	BL	255	OCD
23	DRAWB	114	STANDARD	188	BOTTOMRIGHT	258	TYPENAME
24	POSITION	115	SCREEN	188	BR	259	FILL
24	POS	121	SOLID	189	TEXTMARGIN	260	EPS
25	SETTEXT	122	DASHED	189	TEXTM	261	TIFF
25	SETT	123	DOTTED	190	TOPRIGHT	262	PCL
26	DRAWGRAPHIC	124	LIGHT	190	TR	263	BMP-WIN
26	DRAWG	125	MEDIUM	191	TOPLEFT	264	BMP-OS2
31	DEFINE	126	BOLD	191	TL	265	PCX
31	DEF	127	XLIGHT	192	POSITIONING	266	GIV
32	ENDDEF	128	DARK	192	POSING	267	JFIF
32	END	129	XDARK	193	PATH	268	WIDTH
33	PLACE	130	SPACED	194	TO	269	DEPTH
51	NOSTORE	132	LOCATION	195	CLOSE	271	ELLIPSE
52	STORE	132	LOC	200	AXIS	274	FILLETS
53	REPLACE	133	TOP	200	AXES	275	MARKER
54	ALL	134	CENTER	201	NON	278	SHORTDASH
55	WARN	135	BOTTOM	202	HORIZONTAL	279	DASHDOT
56	ERROR	136	ACROSS	203	VERTICAL	280	DOUBLEDOT
57	NOSUMMARY	137	BALANCE	205	TRACEALL	281	LONGDASH
57	NOSUMM	138	TATE	210	SOSI	282	DSHDBLDOT
58	SUMMARY	140	LINE	211	NOSOSI	285	CROSS
58	SUMM	141	UNDERLINE	212	COLOR	286	PLUS
60	INCHES	141	UNDER	213	OCA	187	DIAMOND
60	IN	141	UND	214	RGB	288	SQUARE
60	INCH	142	CHAR	215	CMYK	289	SIXSTAR
61	MILLIMETERS	143	HEX	216	HIGHLIGHT	290	EIGHTSTAR
61	MILLIMETER	144	NOUNDERLINE	217	CIELAB	291	FILLDMND
61	MM	144	NOUNDER	218	BLUE	292	FILLSQR
62	PELS	144	NOUND	219	RED	293	DOT
62	PEL	145	ABSOLUTE	220	MAGENTA	294	FILL
63	CPI	145	ABS	221	GREEN	295	DOT01
64	LPI	146	MODERN	222	CYAN	296	DOT02
65	POINTS	146	MOD	223	YELLOW	297	DOT03
66	AUTO	147	COLUMN	224	BLACK	298	DOT04
67	LINESP	147	COL	225	BROWN	299	DOT05
70	SIZE	150	TRACE	226	NONE	300	DOT06
71	OFFSET	151	NOTRACE	227	DEFAULT	301	DOT07
80	DDNAME	160	BORDERWEIGHT	228	COVERAGE	302	DOT08
81	FILETYPE	160	BW	229	RVAL	303	VERTLN
82	SCALE	161	BOTH	230	GVAL	304	HORZLN
83	HEIGHT	162	CIRCLE	231	BVAL	305	BLTR1
84	CHARSET	163	CONNECTION	232	CVAL	306	BLTR2
84	CS	164	CORNERLENGTH	233	MVAL	307	TLBR1
85	CODEPAGE	164	CL	234	YVAL	308	TLBR2
85	CP	165	DIAGONAL	235	KVAL	309	NOFILL
90	LEFT	166	DIAMETER	236	LVAL		
91	RIGHT	168	HALF	237	C1VAL		

Appendix J. Storage Summary

When overlays are used in an actual printing application, they must be stored in the printer. In addition, the printer must store the variable data to be merged with the overlay and any fonts or page segments used by the overlay or variable data.

In unusual cases, the printer storage may not be able to hold all these objects. In this situation, the page is not printed. To print the overlay and the data within the storage available to the printer, the job requirements have to be simplified. One way to simplify the job is to reduce the number of fonts or page segments used by the variable data or overlay. Another way is to change the overlay definition so it calls for less text or shading or for fewer lines or images.

To simplify the overlay, you could just reduce the elements at random until the job runs. This hit-and-miss method, while it might be fast, might also result in your eliminating more than you have to. A more precise method is to calculate the total storage requirements of the job, compare them with the storage capacity of your printer, and change only what you must.

As an alternative to simplifying your overlay, if the overlay requires only extra raster image storage, you may consider purchasing more raster image storage for your printer.

The **CONTROL** command in OGL/370 has a subcommand that produces a statistical summary at the end of the definition listing. This summary information can then be used to determine the storage requirements of the overlay and the requirements of the individual elements of the overlay. An explanation of the procedure for determining storage requirements is contained in *Advanced Function Printing: Diagnosis Guide*., (also called the *System Diagnosis Guide*).

The summary portion of the listing is divided into the following headings:

USED FONT SUMMARY

This information helps you to determine the storage required for the fonts requested and the text to be printed.

PAGE SEGMENT SUMMARY

This section lists the page segments used. If your installation does not have a listing of the sizes of your page segments, see the *System Diagnosis Guide* to estimate the storage requirements.

WHOLE IMAGE SUMMARY

This section contains information that helps you calculate the storage requirements of the raster images produced as a result of processing your overlay definition. Several OGL/370 commands produce raster images. Each raster image is listed, with a descriptive name, in the whole image summary:

- The **DEFINE PATTERN** command produces whole images with the name specified in that command.
- Circles have the name \$CIRCLE.
- Miter or bevel connections have the name \$MIT/BEV.
- Segments of dashed lines have the name \$LINESEG.
- Shaded areas of paths have the name \$SHADING.

Sometimes the optimizer chooses to split or merge raster images. In this case, they appear in the whole image summary with the name \$OPTMZER.

SOLID RULE SUMMARY

This section lists a summary of the solid rules of the overlay. Included in this summary are straight box borders, rules drawn with the **DRAWRULE** command, and path segments which are horizontal

or vertical. The contents of this summary may not match the rules defined by the overlay definition, since OGL/370 merges solid rules, where possible, for more efficient printing.

REPEAT CELL SUMMARY

This section gives the lengths and widths (in pels) of shaded boxes and dotted and dashed rules, all of which OGL/370 processes as images. For boxes, calculate from the SOURCE columns rather than the TARGET columns. Dotted and dashed rules may not match the rules defined by the overlay definition, since OGL/370 merges non-solid rules, where possible, for more efficient printing.

After you have received the summary information, refer to the *System Diagnosis Guide* and calculate the storage needs for the overlay.

Appendix K. Measurement Units Conversion Table

Although OGL/370 does not use more than 2 digits to the right of the decimal point, the table below shows 4 digits. The extra digits may be useful to you for rounding up values.

Note: The pel conversions are based on 240-pels per inch.

FROM:		TO:			
		pels	points	millimeters	inches
pels	1	...	0.3000	0.1059	0.0042
	2	...	0.6000	0.2117	0.0084
	3	...	0.9000	0.3175	0.0125
	4	...	1.2000	0.4234	0.0167
	5	...	1.5000	0.5292	0.0209
	6	...	1.8000	0.6350	0.0250
	7	...	2.1000	0.7409	0.0292
	8	...	2.4000	0.8467	0.0354
	9	...	2.7000	0.9525	0.0375
	10	...	3.0000	1.0584	0.0417
	30	...	9.0000	3.1750	0.1250
	60	...	18.0000	6.3500	0.2500
	90	...	27.0000	9.5250	0.3750
	120	...	36.0000	12.7000	0.5000
	150	...	45.0000	15.8750	0.6250
	180	...	54.0000	19.0500	0.7500
	210	...	63.0000	22.2250	0.8750
240	...	72.0000	25.4000	1.0000	
points	1	3.3333	...	0.3528	0.0139
	2	13.3333	...	1.4111	0.0566
	4	20.0000	...	2.1167	0.0833
	6	23.3333	...	2.4659	0.0972
	7	26.6667	...	2.8222	0.1111
	8	30.0000	...	3.1750	0.1250
	9	33.3333	...	3.5278	0.1389
	10	36.6667	...	3.8806	0.1528
	11	40.0000	...	4.2333	0.1667
	12	46.6667	...	4.9389	0.1954
	14	53.3333	...	5.6444	0.2222
	16	60.0000	...	6.3500	0.2500
	18	66.6667	...	7.0556	0.2777
	20	80.0000	...	8.4667	0.3333
	24	100.0000	...	10.5833	0.4167
30	120.0000	...	12.7000	0.5000	
millimeters	1	9.4488	2.8346	...	0.0394
	2	18.8976	5.6693	...	0.0787
	3	28.3465	8.5039	...	0.1181
	4	37.7953	11.3386	...	0.1575
	5	47.2441	14.1732	...	0.1969
	6	56.6929	17.0078	...	0.2362
	7	66.1417	19.8425	...	0.2756
	8	75.5905	22.6771	...	0.3150
	9	85.0394	25.5118	...	0.3543
	10	94.4882	28.3465	...	0.3937

FROM:		TO:			
		pels	points	millimeters	inches
inches	0.1	24.0000	7.2000	2,5400	...
	0.25	60.0000	18.0000	6.3500	...
	0.5	120.0000	36.0000	12.7000	...
	0.75	180.0000	54.0000	19.0500	...
	1.0	240.0000	72.0000	25.4000	...

Appendix L. Codes and Messages

When OGL/370 is finished processing your overlay, input commands and their associated error messages appear on the system output listing.

Most messages provide enough information to solve the problem without referring to a messages and code manual. However, some messages may require a more complete description than the one given in the message text. Therefore, the message listings include:

- Message text
- Insert explanations
- System action, when message text does not provide
- User response.

Note: In EXTENT warnings, the unit of measurement is from the most recent POSITION command.

Codes

The following return codes are set by OGL/370 and returned to the user through the system message which indicates job completion. For all return codes through 8, the sample overlay is always generated and available for printing. For return code 12, the sample overlay will generally be created and available for printing. For return code 16, the sample overlay is not generated.

Following are the generated return codes and their descriptions:

RC=0 Job completed successfully with no errors.

RC=4 Job completed successfully with information messages. The overlay is complete and correct. The informational messages indicate processes performed by OGL during the compilation.

RC=8 Job completed successfully with warning messages. The overlay is complete, but some part of the overlay, such as some text that could not be printed, is indicated in the job listing. These warnings indicate that the overlay may not be satisfactory for a final copy and the source should be corrected and re-compiled.

RC=12 OGL completed the overlay compilation, but parts of the overlay may not be printable due to a resource, such as a font, not being available to OGL at compile time. To make the overlay acceptable, you must correct the failures indicated by the error messages printed in the compiled listing.

RC=16 OGL has encountered a terminating condition which will not allow continued processing. An example would be if the message modules for the language requested when the overlay job was submitted could not be loaded by the operating systems. Other examples include I/O failures when attempting to write to the output data sets (or read from the input data set), insufficient storage for OGL to create its internal data and data buffers (usually an abend), or other abending conditions.

Messages

DZI01011 SYNTAX - ERROR:
THE ENTRY *entry* ON LINE *nn* IS NOT ALLOWED IN THIS POSITION. ENTRIES ALLOWABLE IN THIS POSITION ARE:
KEYWORDS: (*keylist* | NONE)
VALUES: (*desclist* | NONE)
SEE THE SYNTAX RULES FOR THE *command* COMMAND IN THE REFERENCE MANUAL.
THE COMMAND CONTAINING THIS ENTRY WAS NOT PROCESSED.
PROCESSING CONTINUES.

Insert Explanation:

entry Entry specified.
nn Line number.
keylist A list of keywords.
desclist A list of descriptive names for valid entries.
command Command name specified.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0102I SYNTAX - ERROR:
THE ENTRY *entry* ON LINE *nn* IS NOT ALLOWED IN THIS POSITION. AT THIS POSITION A SEMICOLON MAY BE SPECIFIED TO END THE COMMAND OR ONE OF THE FOLLOWING MAY BE SPECIFIED:
KEYWORDS: (*keylist* | NONE)
VALUES: (*desclist* | NONE)
SEE THE SYNTAX RULES FOR THE *command* COMMAND IN THE REFERENCE MANUAL.
THE COMMAND CONTAINING THIS ENTRY WAS NOT PROCESSED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
entry Entry name specified.
keylist A list of keywords.
desclist A list of descriptive names for valid entries.
command Command name specified.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0103I SYNTAX - ERROR:
THE COMMAND BEGINNING ON LINE *nn* IS NOT COMPLETE AND WAS NOT PROCESSED.
SPECIFY ONE OF THE FOLLOWING:
KEYWORDS: (*keylist* | NONE)
VALUES: (*desclist* | NONE)
SEE THE SYNTAX RULES FOR THE *command* COMMAND IN THE REFERENCE MANUAL.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
keylist A list of keywords.
desclist A list of descriptive names for valid entries.
command Command name specified.

User Response: Correct the error and resubmit the job.

DZI0104I SYNTAX - INFORMATIONAL MESSAGE:
THE VALUE SPECIFIED FOR *desc* ON LINE *nn* HAS BEEN TRUNCATED TO TWO DECIMAL PLACES — *value*.
PROCESSING CONTINUES.

Insert Explanation:

desc Descriptive name for entry specified (ex: RULE THICKNESS).
nn Line number.
value Truncated number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0106I **DECIMAL - ERROR:**
THE ENTRY *entry* BEGINNING ON LINE *nn* IS NOT A VALID DECIMAL NUMBER. THE PROBLEM IS ONE OF THE FOLLOWING:
1) THERE ARE MORE THAN FOUR DIGITS TO THE LEFT OF THE DECIMAL POINT
2) THERE IS AN INVALID CHARACTER IN THE NUMBER
3) THERE IS MORE THAN ONE DECIMAL POINT PROCESSING CONTINUES.

Insert Explanation:

entry Invalid string specified when a decimal number expected.
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0107I **PARSE - WARNING:**
THE COMMAND BEGINNING ON LINE *nn* WAS NOT CORRECTLY ENDED BEFORE THE END OF THE COMMAND STREAM WAS REACHED. AN ENDING SEMICOLON IS ASSUMED AT END OF THE COMMAND STREAM.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0109I **PARSE - WARNING:**
THE COMMAND CONTAINING THE *construct* BEGINNING IN LINE *nn* WAS NOT CORRECTLY ENDED BEFORE THE END OF THE COMMAND STREAM WAS REACHED. AN ENDING APOSTROPHE AND SEMICOLON IS ASSUMED AT THE END OF THE COMMAND STREAM. ALL ENTRIES BETWEEN THE BEGINNING APOSTROPHE AND THE ASSUMED APOSTROPHE AND SEMICOLON HAVE BEEN PROCESSED AS A *construct*.
PROCESSING CONTINUES.

Insert Explanation:

construct TEXT STRING or BLOCK COMMENT.
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0111I **PARENTHESIS - WARNING:**
THE LEFT PARENTHESIS ON LINE *nn* DOES NOT HAVE A MATCHING RIGHT PARENTHESIS. A SEMICOLON OR A LEFT PARENTHESIS WAS ENCOUNTERED BEFORE A RIGHT PARENTHESIS WAS ENCOUNTERED. A RIGHT PARENTHESIS IS ASSUMED AT THIS POINT.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0112I **PARSE - WARNING:**
THE *construct* BEGINNING ON LINE *nn* CONTAINS A SEMICOLON THAT IS NOT PART OF A SEMICOLON PAIR. THE *construct* AND THE CONTAINING COMMAND ARE ENDED AT THIS POINT. A SINGLE SEMICOLON SIGNIFIES THE END OF THE COMMAND. TO SPECIFY A SEMICOLON IN THE *construct* TWO SEMICOLONS IN A ROW MUST BE SPECIFIED.
PROCESSING CONTINUES.

Insert Explanation:

construct EXT STRING or BLOCK COMMENT.
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI02011 MISSING COMMAND - ERROR:
THE COMMAND BEGINNING ON LINE *nn* IS AN ACTION COMMAND BUT A VALID OVERLAY COMMAND HAS NOT YET BEEN SPECIFIED. THIS AND ALL SUBSEQUENT COMMANDS WILL ONLY BE CHECKED FOR SYNTAX ERRORS. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI02021 COMMAND SEQUENCE - ERROR:
THE COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. THE *command* COMMAND WAS SPECIFIED OUT OF SEQUENCE. IT MUST BE USED PRIOR TO THE FIRST ACTION COMMAND. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
command ORIENT or CONTROL.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI02031 DUPLICATE COMMAND - ERROR:
THE COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. A VALID *command* COMMAND WAS PREVIOUSLY SPECIFIED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number
command OVERLAY, ORIENT, or CONTROL.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI02041 INVALID COMMAND - ERROR:
THE ENTRY *entry* ON LINE *nn*, WAS NOT A VALID COMMAND NAME. ENTRIES FOUND UP TO THE NEXT SEMICOLON WERE NOT PROCESSED. THE FIRST ENTRY IN A COMMAND LINE (EXCLUDING COMMENTS) MUST BE A VALID COMMAND NAME. PROCESSING CONTINUES.

Insert Explanation:

entry Entry specified.
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0205I **INVALID COMMAND - ERROR:**
NO VALID COMMAND NAME WAS FOUND ON LINE *nn* BEFORE THE ENDING SEMICOLON. THIS LINE WAS NOT PROCESSED. THE FIRST ENTRY IN A COMMAND LINE (EXCLUDING COMMENTS) MUST BE A VALID COMMAND NAME.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0206I **OVERLAY - ERROR:**
THE OVERLAY CANNOT BE STORED USING THE NAME SPECIFIED IN THE OVERLAY COMMAND. THE NAME CONTAINED AT LEAST ONE CHARACTER WHICH IS NOT ALLOWED. ALLOWABLE CHARACTERS ARE A-Z, 0-9, @, \$ AND #.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0301I **NOTIFY - ERROR:**
NOTIFY THE SYSTEM PROGRAMMER.
AN ERROR OPENING AN *type* DATA SET WITH A DDNAME OF *ddname* HAS BEEN DETECTED.
PROCESSING CONTINUES.

Insert Explanation:
type INPUT or OUTPUT.
ddname DDNAME specified.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0302I **NOTIFY - ERROR:**
NOTIFY THE SYSTEM PROGRAMMER.
THE CONTROL COMMAND WAS SPECIFIED WITH THE STORE OR REPLACE OPTION, HOWEVER JOB CONTROL INFORMATION IS MISSING FOR THE OVERLAY LIBRARY. THE OVERLAY IS NOT STORED IN THE OVERLAY LIBRARY.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0303I **NOTIFY - ERROR:**
NOTIFY THE SYSTEM PROGRAMMER.
AN ERROR WAS DETECTED IN READING THE JFCB WITH DDNAME *ddname*. RDJFCB RETURN CODE = *rc*.
PROCESSING CONTINUES.

Insert Explanation:
ddname DDNAME that encountered error.
rc Return code issued by RDJFCB macro.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0304I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
AN ERROR OPENING THE SYSPRINT DATASET HAS BEEN DETECTED.
CHECK THE JCL TO ENSURE THE DDNAME WAS CORRECTLY SPECIFIED.
PROCESSING STOPS.

Explanation: The text is displayed as a write-to-operator (WTO) message routed for programmer attention. The message is issued by OGL/MVS only.

System Action: Processing stops.

User Response: Notify the system programmer.

DZI0305I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE MEMBER *memname* WAS NOT FOUND IN THE *dsname* DATA SET. THE CORRESPONDING
DDNAME IS *ddname*. ENSURE THAT THE MEMBER NAME WAS CORRECTLY SPELLED.
PROCESSING CONTINUES.

Insert Explanation:

memname Member name that was not found.
dsname Data set name in which the member was not found.
ddname DDNAME that encountered the error.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0306I STORE - ERROR:
THE CONTROL COMMAND SPECIFIED THE STORE OPTION, BUT THE OVERLAY LIBRARY
MEMBER, *memname*, ALREADY EXISTS. THE OVERLAY WAS NOT STORED. IF YOU WISH TO
REPLACE THIS MEMBER, SPECIFY THE REPLACE OPTION ON YOUR CONTROL COMMAND.
PROCESSING CONTINUES.

Insert Explanation:

memname Member name.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0307I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE DDNAME *ddname* SPECIFIED A MEMBER NAME *memname*. A MEMBER NAME MAY NOT BE
SPECIFIED FOR A DATA SET IDENTIFIED BY DDNAME *ddname*. ANY DATA SET IDENTIFIED BY
THIS DDNAME WILL NOT BE USED. ACCESS TO THIS DATA SET IS NOT ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

ddname DDNAME that encountered error.
memname Member name that was not found.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0309I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE DDNAME *ddname* SPECIFIED A DISPOSITION OF SHR. A DISPOSITION OF NEW OR OLD IS
ALLOWABLE FOR THIS DDNAME. ANY DATA SET IDENTIFIED BY THIS DDNAME WILL NOT BE
USED. ACCESS TO THIS DATA SET IS NOT ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

ddname DDNAME that encountered error.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0310I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE DDNAME *ddname* SPECIFIED CONCATENATION. CONCATENATION IS NOT ALLOWED FOR THIS DDNAME. ANY DATA SET IDENTIFIED BY THIS DDNAME WILL NOT BE USED. ACCESS TO THIS DATA SET IS NOT ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

ddname DDNAME that encountered error.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0311I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE DCB SYNAD EXIT HAS BEEN INVOKED WHILE PROCESSING THE DDNAME DESCRIBED BELOW. THE PERTINENT I/O INFORMATION IS AS FOLLOWS:
**** DDNAME: *ddname***
**** OPERATION ATTEMPTED: <mv>op:emv.**
**** ERROR DESCRIPTION: *desc***
ACCESS TO THE DATA SET REFERENCED BY THIS DDNAME IS SUSPENDED.
PROCESSING CONTINUES.

Insert Explanation:

ddname DDNAME that encountered error.

op GET, PUT, OPEN, or CLOSE.

desc Description of the error.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0312I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE DCB ABEND EXIT HAS BEEN INVOKED WHILE PROCESSING THE DDNAME DESCRIBED BELOW. THE PERTINENT I/O INFORMATION IS AS FOLLOWS:
**** DDNAME: *ddname***
**** OPERATION ATTEMPTED: *op***
**** SYSTEM COMPLETION CODE: *code1***
**** SYSTEM RETURN CODE: *code2***
ACCESS TO THE DATA SET REFERENCED BY THIS DDNAME IS SUSPENDED. SEE THE "OS/V5 MESSAGES LIBRARY: VS2 SYSTEM CODES" MANUAL FOR MORE INFORMATION.
PROCESSING CONTINUES.

Insert Explanation:

ddname DDNAME that encountered error.

op GET, PUT, OPEN, or CLOSE.

code1 System completion code.

code2 System return code.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0315I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE SEGMENT *segname* WAS NOT FOUND IN THE SEGMENT DATA SET. THE CORRESPONDING
DDNAME IS *ddname*.
PROCESSING CONTINUES.

Insert Explanation:

segname Segment name that was not found.
ddname DDNAME that encountered the error.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0316I JCL PARSE - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
MORE THAN *parmnum* PARAMETERS WERE PASSED TO THE OVERLAY GENERATION
LANGUAGE PROGRAM. ONLY *parmnum* ARE ALLOWED. CORRECT THE PARAMETER LIST AND
RESUBMIT THE JOB.
THE OVERLAY GENERATION LANGUAGE COMMAND STREAM WILL NOT BE PROCESSED.

Insert Explanation:

parmnum
Number of parameters allowed.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0317I JCL PARSE - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE JCL PARAMETER LIST CONTAINS MORE THAN ONE LANGUAGE SPECIFICATION.
CORRECT THE PARAMETER LIST AND RESUBMIT THE JOB.
THE OVERLAY GENERATION LANGUAGE COMMAND STREAM WILL NOT BE PROCESSED.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0318I JCL PARSE - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE JCL PARAMETER LIST CONTAINS MORE THAN ONE PARAMETER INDICATING THE
PRESENCE OR ABSENCE OF SEQUENCE NUMBERS. CORRECT THE PARAMETER LIST AND
RESUBMIT THE JOB.
THE OVERLAY GENERATION LANGUAGE COMMAND STREAM WILL NOT BE PROCESSED.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0319I CSECT - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE *type* CSECT FOR THE LANGUAGE "*lan*" CANNOT BE LOADED.
THE OVERLAY GENERATION LANGUAGE COMMAND STREAM WILL NOT BE PROCESSED.

Explanation: A CSECT for the requested language could not be found. One of the following has occurred:

- The language code you have specified is incorrect.
- The language code you have specified is not available at your site.
- The CSECT is corrupted or has been incorrectly installed.

Insert Explanation:

type Either MESSAGE, INSERT or KEYWORD.
lan Language code.

System Action: Processing terminates.

User Response: Check the language that you have requested in your invocation of OGL/370. If you have specified the language incorrectly, correct the error and resubmit the job. If the language is correct, contact the system programmer.

DZI0320I **NOTIFY - WARNING:**
THE *ds* DATASET DOES NOT CONTAIN RECORD LENGTH AND BLOCKSIZE VALUES.
***ds* RECORD LENGTH OF *n* CHARACTERS AND BLOCKSIZE OF *m* CHARACTERS USED.**
PROCESSING CONTINUES.

Explanation: The message is issued because one of the following occurred:

- Neither the OVLRLIB nor the SAMPLE library was allocated with block size and record length values. In this case the block size of 8209 and record length of 8205 were used.
- The OVLRLIB was not allocated with block size and record length values, but the SAMPLE library was allocated with these values. In this case the values allocated for the SAMPLE library are used for the OVLRLIB library.
- The SAMPLE library was not allocated with block size and record length values, but the OVLRLIB library was allocated with these values. In this case the values allocated for the OVLRLIB library are used for the SAMPLE library.

Insert Explanation:

n The length of the record in characters.
m The length of the block in characters.
ds The dataset name (OVLRLIB or SAMPLE).

System Action: Processing continues

User Response: Re-allocate the dataset if the assigned block size and record length values are not as required, using the desired values, or zeros, if values are to be assigned through a DD statement. Re-run the job to recreate the overlay if different record size is necessary.

DZI0321I **NOTIFY - WARNING:**
NOTIFY THE SYSTEM PROGRAMMER.
AN ATTEMPT WAS MADE TO ALTER EXISTING RECORD LENGTH AND BLOCKSIZE VALUES FOR
THE *ds* DATASET. THE *type* WILL NOT BE STORED IN THIS DATASET.
PROCESSING CONTINUES.

Explanation: The library has been previously assigned block size and logical record lengths, and the DD statement contains different values. This is not allowed, as it would lead to library corruption, if members exist in the library. Overlays or samples are not stored.

Insert Explanation:

ds The dataset name (OVLRLIB or SAMPLE).
type OVERLAY or SAMPLE.

System Action: Processing continues to complete command parsing.

User Response: Correct the DD statement in the JCL and re-submit the job.

DZI0322I **NOTIFY - WARNING:**
NOTIFY THE SYSTEM PROGRAMMER.
THE *type* OF THE SAMPLE AND OVERLAY DATASETS DO NOT MATCH.
THE *size* OF *n* CHARACTERS OF THE *ds* DATA SET WILL BE USED.
PROCESSING CONTINUES.

Explanation: The block sizes (or record lengths) of the OVLRLIB and SAMPLE libraries were found not to match. The smaller value was used when determining the size of record needed to write to either the OVLRLIB or SAMPLE libraries.

Insert Explanation:

type RECORD LENGTHS or BLOCKSIZES.
size SHORTER RECORD LENGTH or SMALLER BLOCKSIZE.
ds The dataset name (OVLRLIB or SAMPLE).
n The length of the record in characters.

System Action: Processing continues.

User Response: The smaller block size (or shorter record length) will be used to write records to the larger dataset. To forestall future issuance of this message, allocate both datasets with the same block size and record length.

DZI0323I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE RECORD LENGTH OF *n* CHARACTERS OF THE *ds* DATASET IS INVALID.
VALID RANGE IS 2048-32756 CHARACTERS.
THE *type* WILL NOT BE STORED IN THIS DATASET.
PROCESSING CONTINUES.

Explanation: Minimum or maximum record length limits have been violated for either the OVRLIB or SAMPLE libraries. The message is also produced if the record length is within the valid range, but the block size is not at least 4 greater than the record length. The minimum allowable value is 2048 bytes. The maximum allowable value is 32756 bytes. The erroneous values could have been specified prior to running OGL/370 or they could have been specified through DD statements.

Insert Explanation:

n The record length in characters.
ds The dataset name (OVRLIB or SAMPLE).
type OVERLAY or SAMPLE.

System Action: Processing continues to complete command parsing.

User Response: If the erroneous values were specified prior to running OGL/370, the dataset must be reallocated with valid values. Using zero values will let OGL/370 assign default values or allow you to specify the values using a DD statement. :p.If the erroneous values were specified via DD statements, then the DD statement must be altered to reflect valid values, or the file could be allocated with these values prior to re-running OGL/370.

DZI0335I FILE NOT FOUND - ERROR:
THE *xxx* FILE WAS NOT FOUND. ENSURE THAT THE FILE ID WAS SPECIFIED CORRECTLY AND THAT THE DISK CONTAINING IT IS ACCESSED.
PROCESSING CONTINUES.

Insert Explanation:

xxx Symbolic or segment.

System Action: Processing continues.

User Response: Correct the file id specified or notify the system programmer.

DZI0336I INVALID STORE OPTION - ERROR:
THE CONTROL COMMAND SPECIFIED THE STORE OPTION, BUT THE OVERLAY FILE, '*fileid*', ALREADY EXISTS. TO REPLACE AN EXISTING OVERLAY FILE, SPECIFY THE REPLACE OPTION IN THE CONTROL COMMAND. THE OVERLAY FILE WAS NOT REPLACED.
PROCESSING CONTINUES.

Insert Explanation:

fileid File id for overlay found.

System Action: Processing continues.

User Response: Change STORE to REPLACE or delete the OVLY38PP file.

DZI0337I PSF NOT AVAILABLE - WARNING:
NOTIFY YOUR SYSTEM PROGRAMMER.
THE PRINT OPTION WAS SPECIFIED BUT PRINT SERVICES FACILITY (PSF/VM) WAS NOT AVAILABLE.
A SAMPLE-OVERLAY FILE WAS CREATED ON YOUR "A" DISK BUT THE FILE WAS NOT PRINTED.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0338I PSF ERROR - WARNING:
FILENAME = *fn*
FILETYPE = *ft*
FILEMODE = *fm*
PSF ERROR CODE = *rc*
NOTIFY YOUR SYSTEM PROGRAMMER.
THE PRINT OPTION WAS SPECIFIED IN THE INVOCATION. THE PRINT SERVICES FACILITY
MODULE, APRLPFC, WAS CALLED TO INVOKE THE PSF COMMAND BUT AN ERROR CODE
WAS RETURNED. THE DESCRIPTION OF THE ERROR CODE MAY BE FOUND IN THE *PSF*
***MESSAGES AND CODES* MANUAL.**
A SAMPLE-OVERLAY FILE WAS CREATED ON YOUR "A" DISK BUT THE FILE WAS NOT
PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

fn Filename of the sample overlay.
ft Filetype of the sample overlay.
fm Filemode of the sample overlay.
rc Error code returned.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI000341I DISK IS FULL - ERROR:
NO MORE SPACE IS AVAILABLE ON YOUR DISK TO COMPLETE PROCESSING.
PROCESSING STOPS.

Explanation: The text is displayed on the screen using the WRTERM macro.

System Action: Processing stops.

User Response: Make more space available on your disk.

DZI000342I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
A SYSTEM ERROR HAS OCCURRED WHILE PROCESSING THE FOLLOWING INFORMATION:
**** MACRO NAME: *macro***
**** FILE NAME: *fn***
**** FILE TYPE: *ft***
**** FILE MODE: *fm***
**** ERROR CONDITION: *error***
ACCESS TO THE FILE IS SUSPENDED. SEE THE *VM/SP PRODUCT: CMS COMMAND AND MACRO*
***REFERENCE* MANUAL FOR MORE INFORMATION.**
PROCESSING CONTINUES.

Insert Explanation:

macro FSOPEN, FSCLOSE, FSREAD, and so forth.
fn Filename being used.
ft Filetype being used.
fm Filemode being used.
error Error condition returned by the macro.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI000343I ERROR OCCURRED WRITING TO *fn ft fm*, THE ERROR CODE FROM THE FSWRITE MACRO IS *rc*, PROCESSING STOPS.

Insert Explanation:

fn Filename being used.
ft Filetype being used.
fm Filemode being used.
rc Error condition returned by the FSWRITE macro.

System Action: Processing stops.

User Response: Notify the system programmer.

DZI000345I AN ASTERISK MAY NOT BE SPECIFIED IN THE FILENAME OR FILETYPE IN THE PROGRAM INVOCATION. PROCESSING STOPS.

System Action: Processing stops.

User Response: Notify the system programmer.

DZI000346I NO INPUT FILENAME WAS SPECIFIED IN THE PROGRAM INVOCATION. REFER TO THE USER'S GUIDE FOR HELP. PROCESSING STOPS.

System Action: Processing stops.

User Response: Notify the system programmer.

DZI000347I THE INPUT FILE, *fn ft fm*, WAS NOT FOUND. CHECK THE FILE ID AND DISK SPECIFICATION. PROCESSING STOPS.

Insert Explanation:

fn Filename requested.
ft Filetype requested.
fm Filemode requested.

System Action: Processing stops.

User Response: Notify the system programmer.

**DZI0348I PSF NOT AVAILABLE - WARNING:
NOTIFY YOUR SYSTEM PROGRAMMER.
THE PRINT OPTION WAS SPECIFIED BUT A DMSFREE OR DMSFRET ERROR OCCURRED WHILE ATTEMPTING TO INVOKE PRINT SERVICES FACILITY (PSF/VM). A SAMPLE-OVERLAY FILE WAS CREATED ON YOUR "A" DISK BUT THE FILE WAS NOT PRINTED.
PROCESSING CONTINUES.**

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0349I FULL SCREEN MODE IS OFF OR SUSPENDED. SET FULL SCREEN ON TO ALLOW TERMINAL MESSAGES TO BE DISPLAYED IN DBCS.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

**DZI0353I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE SEGMENT *segname* WAS NOT FOUND IN THE SEGMENT LIBRARY. THE MEMBER TYPE USED FOR THIS SEGMENT WAS *type*.
PROCESSING CONTINUES.**

Insert Explanation:

segname Segment name.
type Segment type.

System Action: Processing continues.

User Response: Notify the system programmer.

**DZI0354I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE DTF ERROR EXIT HAS BEEN INVOKED WHILE PROCESSING A DTF FOR THE
REFERENCED FILE. THE PERTINENT INFORMATION IS AS FOLLOWS:
** FILE NAME:*name*
** MACRO INVOKED:*op*
** ERROR DESCRIPTION:*desc*
PROCESSING CONTINUES.**

Insert Explanation:

name A logical unit name (SYSIPT or SYSLST) or the file name SYMBOLIC.
op GET, PUT, OPEN, or CLOSE.
desc Description of the error.

System Action: Processing continues.

User Response: Notify the system programmer.

**DZI0355I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
A LIBRARIAN ERROR HAS OCCURRED WHILE PROCESSING THE TYPE LIBRARY REFERENCED
BELOW. THE PERTINENT INFORMATION IS AS FOLLOWS:
** MEMBER NAME: *memname*
** MEMBER TYPE: *memtype*
** MACRO INVOKED: *macro-operand*
** RETURN CODE: *retcode*
** FEEDBACK CODE: *feedcode*
ACCESS TO THE MEMBER IS NOT ESTABLISHED.
PROCESSING CONTINUES.**

Insert Explanation:

memname Member name
memtype Member type
macro-operand VSE librarian macro invoked
retcode Librarian return code.
feed code Librarian feed code.

System Action: Processing continues.

User Response: Notify the system programmer.

**DZI0356I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE "DLBL" JCS STATEMENT FOR FILENAME *filename* IS MISSING OR INVALID. ACCESS TO
THE FILE REQUIRES A "DLBL" STATEMENT FOLLOWED BY AN "EXTENT" STATEMENT. AN
"ASSGN" STATEMENT MUST ALSO BE SPECIFIED IN THE JCS. ACCESS TO THE FILE IS NOT
ESTABLISHED.
PROCESSING CONTINUES.**

Insert Explanation:

filename Filename required.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0357I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE "EXTENT" JCS STATEMENT FOR FILENAME *filename* IS MISSING OR INVALID. ACCESS TO
THE FILE REQUIRES AN "EXTENT" STATEMENT AFTER THE "DLBL". STATEMENT. AN "ASSGN".
STATEMENT MUST ALSO BE SPECIFIED IN THE JCS. ACCESS TO THE FILE IS NOT
ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

filename Filename required.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0358I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE "ASSGN" JCS STATEMENT FOR LOGICAL UNIT SYS*nnn* IS MISSING OR INVALID. ACCESS
TO THE FILENAME *filename* REQUIRES AN "ASSGN" STATEMENT SPECIFYING THIS LOGICAL
UNIT. ACCESS TO THE FILE IS NOT ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

SYS*nnn* Logical unit.

filename Filename required.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0359I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE :q.EXTENT:eq. JCS STATEMENT SPECIFIES A SYSTEM LOGICAL UNIT OF SYS*nnn* ACCESS
TO THE FILENAME *filename* REQUIRES AN "EXTENT" STATEMENT SPECIFYING A
PROGRAMMER LOGICAL UNIT IN THE FORM: SYSNNN (WHERE NNN IS BETWEEN 000 AND
014). ENSURE THAT THE CORRESPONDING "ASSIGN" JCS STATEMENT ALSO SPECIFIES A
PROGRAMMER LOGICAL UNIT WITHIN THIS RANGE. ACCESS TO THE FILE IS NOT
ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

SYS*nnn* Logical unit.

filename Filename.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0360I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE FILE *fileid* SPECIFIED ON THE "DLBL" JCS STATEMENT WITH FILENAME *filename* COULD
NOT BE FOUND ON THE VOLUME *volser*. ACCESS TO THE FILE IS NOT ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

fileid Fileid on DLBL statement.

filename Filename.

volser Volume serial number.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI03611 NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE :q.EXTENT:eq. JCS STATEMENT SPECIFIES A PROGRAMMER LOGICAL UNIT OF SYS nnn .
ACCESS TO THE FILENAME *filename* REQUIRES AN "EXTENT" STATEMENT SPECIFYING A
PROGRAMMER LOGICAL UNIT IN THE FORM: SYSNNN (WHERE NNN IS BETWEEN 000 AND
014). ENSURE THAT THE CORRESPONDING "ASSIGN" JCS STATEMENT ALSO SPECIFIES A
PROGRAMMER LOGICAL UNIT WITHIN THIS RANGE. ACCESS TO THE FILE IS NOT
ESTABLISHED.
PROCESSING CONTINUES.

Insert Explanation:

SYS nnn Logical unit.
filename Filename.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI03711 CSECT - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
type COULD NOT BE LOADED.
THE OVERLAY GENERATION LANGUAGE COMMAND STREAM WILL NOT BE PROCESSED.

Explanation: DZILPARM or a CSECT containing the default language could not be loaded. DZILPARM or the default language has been incorrectly installed, or is corrupted.

Insert Explanation:

type DZILPARM or either default message, insert or keyword csect name.

System Action: Processing terminates.

User Response: Notify the system programmer.

DZI03721 PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE SYMBOLIC KEYWORD WAS FOUND WITH NO LEFT PARENTHESIS FOLLOWING IT FOR
INCLUDING THE SYMBOLIC INFORMATION. THIS KEYWORD IS IGNORED. CORRECT THE
PARAMETER LIST AND REISSUE THE COMMAND.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI03731 PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE SYMBOLIC KEYWORD WAS FOUND WITH NO FILENAME WITHIN THE PARENTHESES. THIS
KEYWORD IS IGNORED. CORRECT THE PARAMETER LIST AND REISSUE THE COMMAND.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI03741 PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
TOO MUCH INFORMATION WAS FOUND WITHIN THE PARENTHESES FOLLOWING THE
SYMBOLIC KEYWORD. THE SYMBOLIC FILEID IS FOUND TO BE *fileid* AND A RIGHT
PARENTHESIS IS ASSUMED FOLLOWING IT. CORRECT THE PARAMETER LIST AND REISSUE
THE COMMAND.
PROCESSING CONTINUES.

Insert Explanation:

fileid Symbolic fileid.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0375I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE PARAMETER LIST CONTAINS MORE THAN ONE PARAMETER INDICATING THE SYMBOLIC
FILE INFORMATION. CORRECT THE PARAMETER LIST AND REISSUE THE COMMAND. THE
FIRST SYMBOLIC SPECIFICATION WILL BE USED.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0376I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE PARAMETER LIST CONTAINS MORE THAN ONE PARAMETER INDICATING WHETHER OR
NOT THE SAMPLE OVERLAY IS TO BE PRINTED. CORRECT THE PARAMETER LIST AND
REISSUE THE COMMAND. THE FIRST SPECIFICATION OF "*sample*" WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:
sample FILE or PRINT.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0377I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE PARAMETER LIST CONTAINS MORE THAN ONE LANGUAGE SPECIFICATION. CORRECT
THE PARAMETER LIST AND REISSUE THE COMMAND. THE FIRST SPECIFICATION OF "*lang*"
WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:
lang ENGLISH or GERMAN.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0378I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE PARAMETER LIST CONTAINS MORE THAN ONE PARAMETER INDICATING THE PRESENCE
OR ABSENCE OF SEQUENCE NUMBERS. CORRECT THE PARAMETER LIST AND REISSUE THE
COMMAND. THE FIRST SPECIFICATION OF "*seq*" WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:
seq NOSEQUEN or SEQUENCE.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0379I SYMBOLIC FILEID NOT SPECIFIED - ERROR:
SYMBOLIC TEXT WAS USED IN THE OVERLAY DEFINITION BUT NO SYMBOLIC FILEID WAS
SPECIFIED IN THE PROGRAM INVOCATION. IF NECESSARY, REFER TO THE OVERLAY
GENERATION LANGUAGE USER'S GUIDE FOR DETAILS ON HOW TO INVOKE THE PROGRAM.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0380I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
NO LEFT PARENTHESIS WAS FOUND FOLLOWING THE FILE IDENTIFIER FOR THE INPUT FILE.
THE INPUT FILE ID BEING USED IS *fn ft fm*. ALL OTHER PARAMETERS ARE IGNORED.
CORRECT THE PARAMETER LIST AND RESUBMIT THE JOB.
PROCESSING CONTINUES.

Insert Explanation:

fn Filename found.
ft Filetype found.
fm Filemode found.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0381I PROGRAM INVOCATION - WARNING
A LEFT PARENTHESIS WAS FOUND WHERE IT WAS NOT NEEDED. IT IS IGNORED. IF YOU
WISH, CORRECT THE PARAMETER LIST AND RESUBMIT THE JOB.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0382I PROGRAM INVOCATION - WARNING:
A RIGHT PARENTHESIS WAS FOUND WHERE IT WAS NOT NEEDED. IT IS IGNORED. IF YOU
WISH, CORRECT THE PARAMETER LIST AND RESUBMIT THE JOB.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0383I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER. THE PARAMETER "*badparm*", PASSED TO THE OVERLAY
GENERATION PROGRAM IS INVALID. CORRECT THE PARAMETER LIST AND REISSUE THE
COMMAND. THIS VALUE IS IGNORED. PROCESSING CONTINUES.

Insert Explanation:

badparm Invalid parameter.

System Action: Processing continues.

User Response: None

DZI0384I PROGRAM INVOCATION - WARNING:
IN THE PROGRAM INVOCATION THE SYMBOLIC FILE ID MAY NOT BE SPECIFIED USING AN
ASTERISK IN THE FILE NAME OR FILE TYPE. THE SYMBOLIC FILE ID SPECIFICATION IS
IGNORED.
PROCESSING CONTINUES.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0385I PROGRAM INVOCATION - ERROR:
NOTIFY THE SYSTEM PROGRAMMER. THE PARAMETER *parm*, PASSED TO THE OVERLAY
GENERATION PROGRAM IS INVALID. CORRECT THE PARAMETER LIST AND REISSUE THE
COMMAND. THIS VALUE IS IGNORED. PROCESSING CONTINUES.

Insert Explanation:

parm Invalid parameter passed to the program on invocation.

System Action: Processing continues without the value.

User Response: Correct the parameter and resubmit the job.

DZI0401I NO UNITS - WARNING:
THE ENTRY FOR *desc* IN THE COMMAND BEGINNING ON LINE *nn* MUST EITHER BE SPECIFIED WITH UNITS OR THE COMMAND MUST BE PRECEDED BY A SETUNITS COMMAND WITH NON ZERO VALUES. *desc* IS SET TO ZERO.
PROCESSING CONTINUES.

Explanation:

Insert Explanation:

desc Descriptive name for entry specified (ex: BOX LENGTH).

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0402I PATTERN - ERROR:
IN THE PATTERN DEFINITION BEGINNING ON LINE <mv>*nn*:emv., THE FOLLOWING PATTERN LINES EACH CONTAINED ONE OR MORE RUN LENGTH VALUES THAT EXCEED THE MAXIMUM ALLOWED: *list*. CHECK THE COMMAND TO ENSURE THAT "type" IS THE CORRECT KEYWORD AND THAT ALL RUN LENGTH VALUES ARE SEPARATED BY BLANKS.
THE PATTERN DEFINITION IS ACCEPTED BUT THE LINES LISTED ABOVE WILL CONTAIN ALL "OFF" PELS.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

list List of pattern line numbers.

type Keyword ENCODED.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0403I MAX VALUE EXCEEDED - WARNING:
THE ENTRY FOR *desc* IN THE COMMAND BEGINNING ON LINE *nn*, WITH EITHER THE SPECIFIED UNITS OR THE CURRENT SETUNITS VALUE, HAS EXCEEDED THE LARGEST VALUE THAT CAN BE ACCEPTED. *desc* IS SET TO ZERO.
PROCESSING CONTINUES.

Insert Explanation:

desc Descriptive name for the entry specified (ex: BOX LENGTH).

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0404I PATTERN - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A PEL PATTERN WITH *value* INVALID CHARACTER(S) ON THE *n*th PATTERN LINE. THE PEL PATTERN MUST CONTAIN ONLY 0'S AND 1'S. ANY INVALID CHARACTERS WERE SKIPPED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

value Number of invalid characters.

*n*th Pattern line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0405I **PATTERN - WARNING:**
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A PEL PATTERN WITH *value* INVALID CHARACTER(S) ON THE *n*th PATTERN LINE. THE PEL PATTERN MUST CONTAIN ONLY NUMBERS. ANY INVALID ENTRIES WERE SKIPPED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
value Number of invalid characters.
*n*th Pattern line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0406I **EXTENT - WARNING:**
IN THE DRAWRULE COMMAND BEGINNING ON LINE *nn*, THE *n*th RULE EXTENDS PAST THE *direction* OVERLAY BOUNDARY BY *value unit*. THE PORTION OF THE RULE OUTSIDE THE OVERLAY WILL NOT BE PRINTED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th The *n*th rule number.
direction LEFT, TOP, RIGHT or BOTTOM.
value Amount of error.
unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0407I **EXTENT - WARNING:**
IN THE DRAWRULE COMMAND BEGINNING ON LINE *nn*, THE *n*th RULE, AS SPECIFIED, WOULD BE ENTIRELY OUTSIDE THE *direction* OVERLAY BOUNDARY BY *value unit*. THIS RULE WILL NOT BE PRINTED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th The *n*th rule number.
direction LEFT, TOP, RIGHT or BOTTOM.
value Amount of error.
unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0408I **NOTIFY - WARNING:**
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIES TEXT WHICH CONTAINS THE SYMBOLIC *symbol*. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND. THE REPLACEMENT TEXT FOR THIS SYMBOLIC WAS IN HEXADECIMAL AND CONTAINED EITHER AN ODD NUMBER OF CHARACTERS OR A CHARACTER THAT WAS NOT A VALID HEXADECIMAL CHARACTER. THE RESULTS OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT ARE UNPREDICTABLE. NOTIFY THE SYSTEM PROGRAMMER. PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
symbol Symbol name.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0409I *command* - **WARNING:**
**THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIES A *dimension* TOO SMALL TO DRAW A *type figure*. THIS LINE TYPE WAS CHANGED TO SOLID. CAREFULLY CHECK THE *dimension* AGAINST THE LINE THICKNESS OF THE *figure*.
PROCESSING CONTINUES.**

Explanation: The box or circle you have attempted to draw did not have enough room for the minimum number of dots or dashes required. These minimums are as follows:

BOXES 2 dots or dashes along a side.
CIRCLES 1 dot or dash per circle quadrant.
DIAGONALS One diagonal: 2 dots or dashes per diagonal.
Two diagonals: 3 dots or dashes in each of the diagonals.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
dimension BOX LENGTH, BOX WIDTH or DIAGONAL LENGTH.
type DOTTED or DASHED.
figure BOX or CIRCLE.

System Action: Processing continues.

User Response: The error may be corrected either by increasing the dimensions to allow more room for the dots or dashes, or by decreasing the line thickness to make the dots or dashes smaller. For example:

- If you specified a DASHED border for a box, either the box length or width (as shown in the message) must be at least six times greater than the border thickness.
- If you specified DOTTED border for a box, either the box length or width (as shown in the message) must be at least twice the border thickness.

DZI0410I **NOTIFY - WARNING:**
**THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES TEXT WHICH CONTAINS THE SYMBOLIC *symbol* ON THE *nth* TEXT LINE. THE REPLACEMENT TEXT FOR THIS SYMBOLIC WAS IN HEXADECIMAL AND CONTAINED EITHER AN ODD NUMBER OF CHARACTERS OR A CHARACTER THAT WAS NOT A VALID HEXADECIMAL CHARACTER. THE RESULTS OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT ARE UNPREDICTABLE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.**

Insert Explanation:

nn Line Number
symbol Symbol Name
nth Number of line text

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0415I **PATTERN - INFORMATIONAL MESSAGE:**
THE PATTERN *pattern*, AS SPECIFIED, IS *width1* PELS WIDE BY *height1* PELS HIGH. THE PATTERN SIZE (A MULTIPLE OF EIGHT PELS IN EACH DIRECTION) IS *width2* PELS WIDE BY *height2* PELS HIGH.

Explanation: This message is issued for every defined pattern regardless of its dimensions. For processing, each

dimension must be a multiple of 8 and is padded with blank pels to achieve this requirement. This does not affect the “look” of the pattern but may affect boundary checking when it is placed near the edge of the overlay.

Insert Explanation:

pattern Pattern name.

width1 Width of the pattern definition, in pels, as specified in the source input stream.

height1 Height of the pattern definition, in pels, as specified in the source input stream.

width2 Width of the pattern after the program has redefined the pattern width to be a multiple of eight.

height2 Height of the pattern after the program has redefined the pattern height to be a multiple of eight.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0416I **EXTENT - WARNING:**
IN THE DRAWBOX COMMAND BEGINNING ON LINE *nn*, THE *n*th BOX EXTENDS PAST THE *direction* OVERLAY BOUNDARY BY *value unit*. THE PORTION OF THE BOX OUTSIDE THE OVERLAY WILL NOT BE PRINTED. ANY TEXT SPECIFIED FOR THIS BOX WILL NOT BE PROCESSED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

*n*th The *n*th box number.

direction LEFT, TOP, RIGHT or BOTTOM.

value Amount of error.

unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0417I **EXTENT - WARNING:**
IN THE DRAWBOX COMMAND BEGINNING ON LINE *nn*, THE *n*th BOX, AS SPECIFIED, WOULD BE ENTIRELY OUTSIDE THE *direction* OVERLAY BOUNDARY BY *value unit*, MEASURED FROM THE TOP LEFT CORNER OF THE BOX. THIS BOX WILL NOT BE PRINTED. ANY TEXT SPECIFIED FOR THIS BOX WILL NOT BE PROCESSED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

*n*th The *n*th box number.

direction LEFT, TOP, RIGHT or BOTTOM.

value Amount of error.

unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0418I **NOTIFY - WARNING:**
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIES THE DOUBLE-BYTE SYMBOLIC *symbol*. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE *n*th WITHTEXT OF THE COMMAND. THE REPLACEMENT TEXT FOR THIS SYMBOLIC CONTAINED AN ODD NUMBER OF BYTES. THE RESULTS OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT ARE UNPREDICTABLE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.

nn Line number.

symbol The symbol name.

*n*th Number of text line within WITHTEXT.

nth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0419I DRAWRULE - WARNING:
THE DRAWRULE COMMAND BEGINNING ON LINE *nn* SPECIFIED A LINE LENGTH TOO SMALL TO DRAW A *rule-type* RULE. THIS LINE TYPE WAS CHANGED TO SOLID. CAREFULLY CHECK THE LINE LENGTH AGAINST THE LINE THICKNESS OF THE RULE.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

rule-type DASHED or DOTTED.

System Action: Processing continues.

User Response: Compare your overlay's rule-length and rule-thickness specifications with the requirements below, make corrections as necessary, and resubmit the job:

- If you specified a DASHED rule in the overlay definition, its length must be at least seven times greater than its thickness.
 - If you specified a DOTTED rule in the overlay definition, its length must be at least three times greater than its thickness.
-

DZI0420I NOTIFY - WARNING:
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES TEXT WHICH CONTAINS THE DOUBLE-BYTE SYMBOLIC *symbol* ON THE *nth* TEXT LINE. THE REPLACEMENT TEXT FOR THIS SYMBOLIC CONTAINED AN ODD NUMBER OF BYTES. THE RESULTS OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT ARE UNPREDICTABLE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

symbol Symbol name.

nth Number of line text.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0421I NOTIFY - WARNING:
THE COMMAND BEGINNING ON LINE *nn* REFERENCES A COLOR DEFINITION NAMED *name* WHICH CANNOT BE USED FOR NON-SOLID LINE SEGMENTS, ROUNDED CORNERS, DIAGONALS, OR PATTERNS. YOU MUST USE AN OCA COLOR DEFINITION. THE OBJECT WILL BE DONE IN THE DEVICE DEFAULT COLOR. TEXT LINE. THE REPLACEMENT TEXT FOR THIS SYMBOLIC CONTAINED AN ODD NUMBER OF BYTES. THE RESULTS OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT ARE UNPREDICTABLE.

System Action: Processing continues.

DZI0422I UNDEFINED NAME - WARNING:
THE COLOR PARAMETER OF THE COMMAND BEGINNING ON LINE *nn* REFERENCES A DEFINITION NAMED *name* WHICH HAS NOT BEEN PREVIOUSLY DEFINED. THE PARAMETER IS IGNORED.

System Action: Processing continues.

DZI0423I NOTIFY - WARNING:
THE COMMAND BEGINNING ON LINE *nn* REQUESTS COLOR AND SHADE >> ROUNDED. THESE PARAMETERS ARE MUTUALLY EXCLUSIVE. THE COLOR REQUEST IS IGNORED.

System Action: Processing continues.

DZI0425I NOTIFY - WARNING:
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIES THE DOUBLE-BYTE SYMBOLIC *symbol*. THE REPLACEMENT TEXT FOR THE SYMBOLIC IS NOT PROPERLY DELIMITED. THE TEXT LINE CONTAINING THE SYMBOLIC IS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND. THE RESULT OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT IS UNPREDICTABLE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
symbol The symbol name.
*n*th Number of text line within WITHTEXT.
*m*th Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0430I EXTENT - WARNING:
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIES A LINE OF TEXT THAT BEGINS OUTSIDE THE *direction* OVERLAY BOUNDARY BY *value* PELS. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND. NO TEXT IN THIS LINE WAS PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
direction TOP, BOTTOM, LEFT or RIGHT.
value Amount of error.
*n*th
*m*th

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0435I NOTIFY - WARNING:
THE SETTEXT COMMAND BEGINNING ON LINE *nn* REFERENCES THE DOUBLE-BYTE SYMBOLIC *symbol*. THE REPLACEMENT TEXT FOR THE SYMBOLIC IS NOT PROPERLY DELIMITED. THE TEXT LINE CONTAINING THE SYMBOLIC IS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND. THE RESULTS OF TRANSLATING AND PRINTING THE REPLACEMENT TEXT ARE UNPREDICTABLE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
symbol Symbol name.
*n*th Number of text line within command.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0440 **EXTENT - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES A LINE OF TEXT THAT BEGINS OUTSIDE THE *direction* OVERLAY BOUNDARY BY *value* PELS. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND. NO TEXT IN THIS LINE WAS PRINTED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
direction TOP, BOTTOM, LEFT or RIGHT.
value Amount of error.
*n*th Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI04511 **DRAWBOX - WARNING:**
THE DRAWBOX COMMAND BEGINNING ON LINE *nn* SPECIFIED ROUNDED CORNERS. THE CURRENT LENGTH FOR BOX CORNERS, *lenvalue*, IS LARGER THAN THE MAXIMUM ALLOWED FOR THIS BOX, *maxvalue*. *maxvalue* WAS USED. PROCESSING CONTINUES.

Explanation: The maximum corner length allowed is 1/2 the length of the shortest side of the box.

Insert Explanation:

nn Line number.
lenvalue Box corner length defined in SETUNITS command.
maxvalue Largest length allowed for this box.

System Action: Processing continues.

User Response: Compare the box's length and width against the corner length specified in the SETUNITS command and the requirements below, make corrections as necessary, and resubmit the job:

- The length of the corner must not be greater than 1/2 the length of the shorter of the two box sides.
-

DZI04531 **SHADE - WARNING:**
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIED A SHADE AREA OF *area* FOR A *figure* WITH NO DIAGONALS. THE SPECIFICATION WAS IGNORED. PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
area TOP, BOTTOM, LEFT or RIGHT.
figure BOX or CIRCLE.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI04551 **FONT - WARNING:**
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIED A TEXT STRING CONTAINING *text_type* CHARACTERS. NO *text_type* FONT WAS SPECIFIED FOR THIS LINE. THE LINE IN ERROR WAS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND. THE TEXT WAS NOT PRINTED. PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
text_type SBCS or DBCS.
*n*th Number of text line within WITHTEXT.
*m*th Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0456I **FONT - WARNING:**
THE *command* COMMAND BEGINNING ON LINE :mv.nn:emv. SPECIFIED A TEXT STRING CONTAINING AN SO OR SI CHARACTER WITH A SPECIFIED MODE OF SOSI1. NO SBCS FONT WAS SPECIFIED FOR THIS LINE. THE SO OR SI CHARACTER(S) WERE NOT REPLACED WITH A BLANK SPACE. THE LINE IN ERROR WAS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND.
PROCESSING CONTINUES.

Explanation: Mode SOSI1 specifies that SOSI characters are to be replaced by a blank space using the SBCS font.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
*n*th Number of text line within WITHTEXT.
*m*th Number of WITHTEXT within command. :edl.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0457I **FONT - WARNING:**
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIED *font1* AND *font2* AS A FONT PAIR. THESE ARE BOTH *font_type* FONTS. A FONT PAIR MUST BE ONE OF EACH TYPE. ONLY *font1* IS ACCEPTED. THE LINE IN ERROR WAS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND.
PROCESSING CONTINUES.

Note: If a font of the type NOT specified in this pair was specified previously in this text line, it is still in effect.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
font1 First font specified in pair.
font2 Second font specified in pair.
font_type SBCS or DBCS.
*n*th Number of text line within WITHTEXT.
*m*th Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0460I **DRAWCIRCLE - WARNING:**
THE DRAWCIRCLE COMMAND BEGINNING ON LINE *nn* SPECIFIED A RADIUS FOR THE CIRCLE TOO SMALL TO BE DRAWN *circle_type*. THE CIRCLE WAS DRAWN SOLID.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
circle_type DOTTED or DASHED.

System Action: Processing continues.

User Response: Compare your circle radius and border thickness specifications with the requirements below, make corrections as necessary, and resubmit the job:

- If you specified a DASHED border for a circle, the corner radius must be at least 2.6 times the border thickness.
- If you specified a DOTTED border for a circle, the corner radius must be at least 1.3 times the border thickness.

DZI04611 **DRAWCIRCLE - WARNING:**
THE DRAWCIRCLE COMMAND BEGINNING ON LINE *nn* SPECIFIED THE *direction* PORTION OF THE CIRCLE MORE THAN ONCE. THE REDUNDANT SPECIFICATION(S) WERE IGNORED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
direction TOP LEFT, TOP RIGHT, BOTTOM LEFT or BOTTOM RIGHT.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI04641 **DRAWCIRCLE - WARNING:**
THE DRAWCIRCLE COMMAND BEGINNING ON LINE *nn* SPECIFIES A PARTIAL CIRCLE WITH DIAGONALS. DIAGONALS ARE ONLY ALLOWED IN WHOLE CIRCLES. THE DIAGONAL SPECIFICATION WAS IGNORED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI04681 **EXTENT - WARNING:**
IN THE DRAWCIRCLE COMMAND BEGINNING ON LINE *nn*, THE *n*th CIRCLE EXTENDS PAST THE *direction* OVERLAY BOUNDARY BY *value unit*. THE PORTION OF THE CIRCLE OUTSIDE THE OVERLAY WILL NOT BE PRINTED. ANY TEXT SPECIFIED FOR THIS CIRCLE WILL NOT BE PROCESSED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th The *n*th circle number.
direction LEFT, TOP, RIGHT or BOTTOM.
value Amount of error.
unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI04691 **EXTENT - WARNING:**
IN THE DRAWCIRCLE COMMAND BEGINNING ON LINE *nn*, THE *n*th CIRCLE, AS SPECIFIED, WOULD BE ENTIRELY OUTSIDE THE *direction* OVERLAY BOUNDARY BY *value unit* MEASURED FROM THE CENTER OF THE CIRCLE. THIS CIRCLE WILL NOT BE PRINTED. ANY TEXT SPECIFIED FOR THIS CIRCLE WILL NOT BE PROCESSED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th The *n*th circle number.
direction TOP, LEFT, RIGHT or BOTTOM.
value Amount of error.
unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

| **DZI0470I** **COMMENTS - WARNING:**
| **IN THE CONTROL BEGINNING ON LINE *nn*, THE TOTAL AMOUNT OF TEXT SUPPLIED IN THE**
| **COMMENTS STRINGS EXCEEDS THE MAXIMUM LENGTH OF 11. TEXT BEYOND THE MAXIMUM**
| **WILL BE IGNORED.**
| **PROCESSING CONTINUES.**

| **Insert Explanation:**
| *nn* Line number.

| **System Action:** Processing continues.

| **User Response:** Correct the error and resubmit the job.

| **DZI0480I** **PLACE BARCODE - WARNING:**
| **THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A SPECIAL FUNCTION THAT IS INVALID**
| **WITH A SEQUENCED SYMBOL. THE SELECTION IS IGNORED.**

| **Insert Explanation:**
| *nn* Line number.

| **System Action:** Processing continues.

| **User Response:** Correct the error and resubmit the job.

| **DZI0481I** **PLACE BARCODE - WARNING:**
| **THE COMMAND BEGINNING ON LINE *nn* SPECIFIES AN INVALID COMBINATION OF (too**
| **numerous to spell out) AND (too numerous to spell out) THE SELECTIONS ARE IGNORED.**
| **PROCESSING CONTINUES.**

| **Insert Explanation:**
| *nn* Line number.

| **System Action:** Processing continues.

| **User Response:** Correct the error and resubmit the job.

| **DZI0482I** **PLACE BARCODE - WARNING:**
| **THE COMMAND BEGINNING ON LINE *nn* SPECIFIES TWO-DIMENSIONAL PARAMETERS FOR A**
| **SYMBOLGY TYPE THAT DOES NOT MATCH THE DEFINED TYPE. THE SELECTIONS ARE**
| **IGNORED.**
| **PROCESSING CONTINUES.**

| **Insert Explanation:**
| *nn* Line number.

| **System Action:** Processing continues.

| **User Response:** Correct the error and resubmit the job.

| **DZI0483I** **PLACE BARCODE - WARNING:**
| **THE COMMAND BEGINNING ON LINE *nn* SPECIFIES TWO-DIMENSIONAL PARAMETERS, BUT**
| **THE SYMBOLGY IS NOT TWO-DIMENSIONAL. THE SELECTIONS ARE IGNORED.**
| **PROCESSING CONTINUES.**

| **Insert Explanation:**
| *nn* Line number.

| **System Action:** Processing continues.

| **User Response:** Correct the error and resubmit the job.

DZI0484I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A COMBINATION OF NUMBER OF ROWS
AND SIZE OF ROW THAT IS INVALID. THE SELECTION IS IGNORED.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0485I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A TEXT STRING OF LENGTH 11. THIS IS
OUTSIDE THE RANGE FOR THE BARCODE TYPE. THE SELECTION IS ACCEPTED, BUT MAY
CAUSE PRINTER ERRORS.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0486I PLACE BARCODE - ERROR:
THE PLACE BARCODE COMMAND BEGINNING ON LINE *nn* SPECIFIES A BARCODE NAME
barcode name THAT WAS NOT PREVIOUSLY DEFINED IN A DEFINE BARCODE COMMAND.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0487I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A TEXT STRING WITH THE HEX KEY WORD.
THE TEXT STRING CONTAINS AN ODD NUMBER OF CHARACTERS. THE LAST CHARACTER
WILL BE IGNORED. THIS MAY CAUSE PRINTER ERRORS.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0488I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES ASTERISK. THIS IS NOT SUPPORTED BY
THE CHOSEN BARCODE TYPE. THE SELECTION IS ACCEPTED, BUT MAY CAUSE PRINTER
ERRORS.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0489I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A BARCODE WITH HRI, ORIENTED AT *orientation* DEGREES. YOU ARE OVER THE LIMIT OF *limit* UNIQUE COMBINATIONS OF FONT, TEXT FORMAT, AND TEXT ORIENTATION THAT MAY BE USED IN A SINGLE OVERLAY. THE BARCODE WILL BE PRINTED WITH THE DEVICE SELECTED FONT.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0490I FONT ORIENTATION - WARNING:
THE PLACE BARCODE COMMAND BEGINNING ON LINE *nn* SPECIFIES ORIENTATION OF *orientation* DEGREES. FONT *fontname* CANNOT BE USED FOR THIS ORIENTATION. THE PRINTER DEFAULT FONT WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0491I DEFINE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A FONT FOR THE BARCODE HRI. HRI IS NOT SUPPORTED BY THIS SYMBOLOGY. THE FONT SPECIFICATION IS IGNORED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0492I DEFINE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES AN UNKNOWN SYMBOLOGY TYPE. IT IS ACCEPTED, BUT CHECKING WILL NOT OCCUR ON SOME PARAMETERS IN THIS COMMAND AND PLACE BARCODE COMMANDS USING THIS DEFINITION.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0493I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A TEXT STRING THAT CONTAINS A CHARACTER THAT IS NOT VALID FOR THE CHOSEN BARCODE TYPE AND MODIFIER. THE BARCODE WILL STILL BE SHOWN. THIS MAY CAUSE PRINTER ERRORS.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0494I PLACE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A TEXT STRING WITH THE HEX KEY WORD.
THE TEXT STRING CONTAINS A CHARACTER THAT WAS NOT HEXADECIMAL. THE BARCODE
WILL BE SHOWN USING THE CHARACTERS BEFORE THE ONE IN ERROR. THIS MAY CAUSE
PRINTER ERRORS.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0495I PLACE BARCODE - WARNING:
THE BARCODE SPECIFIED IN THE PLACE COMMAND BEGINNING ON LINE *nn* WOULD EXTEND
PAST THE *direction* OVERLAY BOUNDARY BY *value* PELS IF THE SPECIFIED HEIGHT IS USED.
THE DEFAULT HEIGHT IS USED.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0496I RANGE EXCEEDED - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A *x* THAT IS NOT WITHIN THE RANGE *n1* TO
n2. THE DEFAULT WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0497I DEFINE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A BAR CODE HEIGHT THAT IS GREATER
THAN THE MAXIMUM OVERLAY DIMENSION. THE DEVICE DEFAULT HEIGHT WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0498I DEFINE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES A COLOR *color name* THAT WAS NOT
DEFINED AS OCA. THE DEVICE DEFAULT COLOR WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:
nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0499I DEFINE BARCODE - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIES AN UNKNOWN MODIFIER FOR THE SPECIFIED TYPE. IT IS ACCEPTED, BUT CHECKING WILL NOT OCCUR ON SOME PARAMETERS IN THIS COMMAND AND PLACE BARCODE COMMANDS USING THIS DEFINITION.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0500I UNDEFINED FONT - WARNING:
THE DEFINE BARCODE COMMAND BEGINNING ON LINE *nn* SPECIFIES A FONT *font name* THAT WAS NOT PREVIOUSLY DEFINED IN A FONT COMMAND. THE DEVICE DEFAULT FONT WILL BE USED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0501I ORIENT - WARNING:
THE COMMAND BEGINNING ON LINE *nn* SPECIFIED AN INVALID ORIENTATION OF *value*. VALID ORIENTATIONS ARE 0, 90, 180 OR 270. A ZERO DEGREE ORIENTATION IS USED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

value Invalid orientation (degrees assumed).

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0502I DUPLICATE - ERROR:
THE COMMAND BEGINNING ON LINE *nn*, DESCRIBES A *resource* WITH A NAME OF *name*, THAT HAS BEEN PREVIOUSLY USED AS A *resource* NAME. THIS DEFINITION IS NOT ACCEPTED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

resource GROUP, PATTERN, SEGMENT or FONT.

name Duplicate name.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0503I INVALID MASK - ERROR:
THE DRAWMASK COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. A TOTALLY BLACK OVERLAY HAS BEEN SPECIFIED BECAUSE ONE OR BOTH OF THE SPACING INTERVALS IS LESS THAN TWO PELS. THE SMALLEST SPACING INTERVAL ALLOWED IS TWO PELS.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0504I **GROUP - ERROR:**
THE COMMAND BEGINNING ON LINE *nn* IS NOT ALLOWED IN A GROUP DEFINITION AND WAS NOT PROCESSED. ALLOWABLE COMMANDS ARE: SETUNITS, POSITION, DRAWRULE, DRAWBOX, DRAWMASK, DRAWCIRCLE, DRAWPATH, AND SETTEXT.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0505I **GROUP - ERROR:**
NO ENDDDEF WAS SPECIFIED FOR THE GROUP DEFINITION *name*. THE END OF THE COMMAND STREAM WAS REACHED. THE GROUP DEFINITION IS NOT COMPLETE AND CANNOT BE PLACED.
PROCESSING CONTINUES.

Insert Explanation:

name Group name.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0506I **ENDDDEF - WARNING:**
THE ENDDDEF COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. NO GROUP OR PATTERN WAS BEING DEFINED WHEN THE ENDDDEF COMMAND WAS ENCOUNTERED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0507I **OVERLAY - ERROR:**
THE OVERLAY COMMAND BEGINNING ON LINE *nn*, AS SPECIFIED, RESULTS IN AN OVERLAY SIZE OF LESS THAN ONE PEL. THIS IS AN INVALID OVERLAY COMMAND.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0508I **EXTENT - WARNING:**
THE POSITION COMMAND BEGINNING ON LINE *nn* SPECIFIES A LOCATION OUTSIDE THE *direction* BOUNDARY OF THE OVERLAY BY *value unit*.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
direction TOP, LEFT, RIGHT or BOTTOM.
value Amount of error.
unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0509I UNDEFINED NAME - ERROR:
THE PLACE COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. THE *type name* HAS NOT BEEN DEFINED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
type GROUP, PATTERN, or SEGMENT.
name Name of the pattern, group, or segment to be placed.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0510I FRACTION - INFORMATIONAL MESSAGE:
IN THE COMMAND BEGINNING ON LINE *nn* A FRACTIONAL VALUE IS NOT ALLOWED FOR THE *desc*. THE DIGITS AFTER THE DECIMAL POINT WERE IGNORED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
desc Descriptive name for entry specified (ex: LINE THICKNESS).

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0511I GROUP - WARNING:
THE COMMAND BEGINNING ON LINE *nn* IS NOT ALLOWED IN A GROUP DEFINITION. THE COMMAND WAS PROCESSED AS THOUGH IT WAS OUTSIDE THE GROUP.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0512I NOTIFY - WARNING:
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIES THE SYMBOLIC *symbol* FOR WHICH NO REPLACEMENT TEXT WAS FOUND. THE LINE IN ERROR IS THE *nth* TEXT LINE SPECIFIED IN THE *mth* WITHTEXT OF THE COMMAND. NO CHARACTER SUBSTITUTION WAS MADE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Explanation:

Insert Explanation:

nn Line number.
symbol Invalid symbolic name.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0513I UNDEFINED FONT - WARNING:
THE TEXT SPECIFIED IN THE COMMAND BEGINNING ON LINE *nn* USING THE FONT *name* WAS NOT PROCESSED. THE FONT WAS NOT PREVIOUSLY DEFINED IN A FONT COMMAND. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
name Font name specified.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0514I UNDEFINED - WARNING:
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIED SHADING FOR *figure* NUMBER *value* WHICH IS UNDEFINED. THIS *figure* WAS NOT SHADED. PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX, DRAWCIRCLE, or DRAWPATH.
nn Line number.
figure BOX, CIRCLE, or PATH.
value Box, circle, or path number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0515I FONT - WARNING:
THE *desc* PARAMETER IN THE FONT COMMAND BEGINNING ON LINE *nn* IS NOT APPLICABLE TO THE *object* NAMED "*fontname*". THE UNNECESSARY SPECIFICATION IS IGNORED. PROCESSING CONTINUES.

Insert Explanation:

desc SCALE or HEIGHT
nn Line number.
object RASTER FONT
fontname Name of font.

System Action: Processing continues, but the font was not used.

User Response: Add the missing value and resubmit the job.

DZI0516I FONT - WARNING:
THE FONT COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. THE FONT CHARACTER SET IS OUTLINE TECHNOLOGY. THE CODED FONT DOES NOT CONTAIN POINTSIZE AND *desc* WAS NOT PROVIDED ON THE FONT COMMAND. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
desc Description of invalid entry.

System Action: Processing continues, but the font was not used.

User Response: Add the missing value and resubmit the job.

DZI0517I VALUE EXCEEDED - WARNING:
THE ENTRY FOR *desc* IN THE COMMAND BEGINNING ON LINE *nn*, IS OUT OF RANGE. PROCESSING CONTINUES.

Insert Explanation:

desc Description of invalid entry.
nn Line number.

System Action: Processing continues.

User Response: Correct the invalid value and resubmit the job.

DZI0519I NOTIFY - ERROR:
THE FONT COMMAND ON LINE *nn* SPECIFIES A FONT FOR WHICH NO INFORMATION WAS FOUND FOR ANY SUPPORTED ORIENTATION. THE FONT NAME SPECIFIED WAS *member*; THE DDNAME USED WAS *ddname*. ENSURE THAT THE FONT NAME WAS SPECIFIED CORRECTLY. FOR ADDITIONAL HELP, NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
member Member ID specified.
ddname DDNAME used.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0520I UNDEFINED - WARNING:
THE *command* COMMAND BEGINNING ON LINE *nn* SPECIFIED TEXT FOR *figure* NUMBER *value* WHICH IS UNDEFINED. THIS TEXT WAS NOT PRINTED.
PROCESSING CONTINUES.

Explanation: You attempted to place text in a circle or box that does not exist. If you have only one box (or circle) the figure number should be "1" for the original figure, or "2" for the first repetition.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
figure BOX or CIRCLE.
value Box or circle number.

System Action: Processing continues.

User Response: Check the number of the box or circle in which you wanted to place text, remembering that the first repetition is the second figure.

DZI0522I NOTIFY - WARNING:
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES THE SYMBOLIC *symbol* FOR WHICH NO REPLACEMENT TEXT WAS FOUND. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND. NO CHARACTER SUBSTITUTION WAS MADE.
NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
symbol Invalid symbolic name.
*n*th Number of text line within command.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0523I PLACE - ERROR:
THE PATTERN SPECIFIED IN THE PLACE COMMAND BEGINNING ON LINE *nn* EXTENDS PAST THE *direction* OVERLAY BOUNDARY BY *value* PELS. THE PATTERN IS NOT PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
direction LEFT, TOP, RIGHT or BOTTOM.
value Amount of error.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0524I PLACE - ERROR:
THE PLACE COMMAND BEGINNING ON LINE *nn* WAS SPECIFIED WHILE THE CURRENT POSITION WAS OUTSIDE THE OVERLAY. THE *type* WAS NOT PLACED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

type PATTERN or SEGMENT.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0525I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE FONT COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. THE RECORD(S) OF THE CORRESPONDING FONT OBJECT ARE EITHER NOT USABLE OR MISSING. THE PERTINENT I/O INFORMATION IS AS FOLLOWS:
** MEMBER NAME: *mname*
** RECORD TYPE(S): *types*
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

mname Font member name.

type Acronym for missing structured field.

System Action: Processing continues.

User Response: Correct the font and resubmit the job.

DZI0526I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE FONT COMMAND BEGINNING ON LINE *nn* WAS NOT PROCESSED. THE RECORD *rectype* IN THE FONT "*name*" LIBRARY IS NOT USABLE OR MAY BE MISSING.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

rectype Record type in font library.

name Font name specified.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0528I SYMBOLIC - WARNING:
THE SYMBOLIC DATA SET OR FILE COULD NOT BE OPENED. NO CHARACTER SUBSTITUTION WAS MADE FOR THE SYMBOLIC "symbol." NO MORE SYMBOLICS WILL BE PROCESSED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

symbol Symbol name.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0529I **INTERIOR SPACE - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE** *nn* **SPECIFIES** *figure* **DIMENSIONS SUCH THAT**
THE *figure* **HAS NO INTERIOR SPACE. NO** *attr* **SPECIFIED IN THIS COMMAND WILL BE**
PROCESSED.
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
figure BOX or CIRCLE.
attr TEXT or SHADING.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0530I **EXTENT - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE** *nn* **SPECIFIES A LINE OF TEXT THAT EXTENDS**
PAST THE *direction* **OVERLAY BOUNDARY BY** *value* **PELS. THE LINE IN ERROR IS THE** *nth* **TEXT**
LINE SPECIFIED IN THE *mth* **WITHTEXT OF THE COMMAND. NO TEXT IN THIS LINE WAS**
PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
direction TOP, BOTTOM, LEFT or RIGHT.
value Amount of error.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0531I **EXTENT - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE** *nn* **SPECIFIES TEXT THAT DOES NOT FIT WITHIN**
THE TEXT MARGIN. THE ERROR WAS IN THE *direction* **DIRECTION. THE TEXT IN ERROR WAS**
SPECIFIED IN THE *mth* **WITHTEXT OF THE COMMAND. THE TEXT WAS PRINTED.**
PROCESSING CONTINUES.

Explanation: If the direction of error is *horizontal* for MODERN text, one or more lines of text are too long using the specified font. If the error is *vertical* for MODERN text, there are too many lines of text using the specified font. Similarly, for *horizontal* and COLUMN/TATE, too many lines; for *vertical* and COLUMN/TATE, too long a line.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
direction HORIZONTAL or VERTICAL.
nth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0532I **UNDERSCORE - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE** *nn* **SPECIFIES FONT** *name* **FOR UNDERLINED**
TEXT. THIS FONT CANNOT BE USED FOR UNDERLINED TEXT. THE LINE IN ERROR IS THE *nth* **TEXT**
LINE SPECIFIED IN THE *mth* **WITHTEXT OF THE COMMAND. THE TEXT IN THIS LINE**
USING FONT *name* **WAS PRINTED BUT NOT UNDERLINED.**
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.

nn Line number.
name FONT name specified.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0533I FONT ORIENTATION - WARNING:
THE *command* **COMMAND BEGINNING ON LINE *nn* SPECIFIES FONT *name* FOR TEXT IN THE *format* **FORMAT, ORIENTED AT *degrees* DEGREES. THIS FONT CANNOT BE USED FOR THAT COMBINATION OF FORMAT AND ORIENTATION. THE LINE IN ERROR IS THE *nth* TEXT LINE SPECIFIED IN THE *mth* WITHTEXT OF THE COMMAND. THE TEXT USING THIS FONT WAS NOT PRINTED.**
PROCESSING CONTINUES.**

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
name FONT name specified.
format MODERN, COLUMN or TATE.
degrees Net orientation.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0534I UNDERSCORE - WARNING:
THE *command* **COMMAND BEGINNING ON LINE *nn* SPECIFIES THAT TEXT IN THE *format* **FORMAT IS TO BE UNDERLINED. TEXT IN EITHER THE COLUMN OR TATE FORMAT CANNOT BE UNDERLINED. THE LINE IN ERROR IS THE *nth* TEXT LINE SPECIFIED IN THE *mth* WITHTEXT OF THE COMMAND. THE TEXT WAS PRINTED BUT NOT UNDERLINED.**
PROCESSING CONTINUES.**

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
format COLUMN or TATE.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0535I DOUBLE-BYTE FONT - WARNING:
THE *command* **COMMAND BEGINNING ON LINE *nn* SPECIFIES AN ODD NUMBER OF BYTES TO BE PRINTED WITH FONT *name* WHICH IS A DOUBLE-BYTE FONT. THE TEXT IN ERROR IS SPECIFIED IN THE *nth* TEXT LINE OF THE *mth* WITHTEXT OF THE COMMAND. THE RESULTS OF TRANSLATING AND PRINTING THE TEXT ARE UNPREDICTABLE.**
PROCESSING CONTINUES.

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
name FONT name specified.
nth Number of text line within WITHTEXT.
mth Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0536I **TOO MANY FONTS - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE *nn* SPECIFIES FONT *name* FOR TEXT IN THE *format* **FORMAT, ORIENTED AT *degrees* DEGREES. YOU ARE OVER THE LIMIT OF *value* UNIQUE COMBINATIONS OF FONT, TEXT FORMAT AND TEXT ORIENTATION THAT MAY BE USED IN A SINGLE OVERLAY. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND. THE TEXT IN THIS LINE USING FONT *name* WAS NOT PRINTED. PROCESSING CONTINUES.****

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
name FONT name specified.
format MODERN, COLUMN, or TATE.
degrees Net text orientation.
value Number of allowable fonts.
*n*th Number of text line within WITHTEXT.
*m*th Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0537I **EXTENT - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE *nn* SPECIFIES BALANCED TEXT THAT DOES NOT FIT INTO THE *figure*. THE TEXT IN ERROR WAS SPECIFIED IN THE *n*th WITHTEXT OF THE COMMAND. THE TEXT WAS PRINTED. PROCESSING CONTINUES.**

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
figure BOX or CIRCLE.
*n*th Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0538I **HEXADECIMAL - WARNING:**
THE *command* **COMMAND BEGINNING ON LINE *nn* SPECIFIED A TEXT STRING WITH THE HEX KEYWORD. THE TEXT STRING EITHER CONTAINED AN ODD NUMBER OF CHARACTERS OR CONTAINED A CHARACTER THAT WAS NOT HEXADECIMAL. THE RESULTS OF TRANSLATING AND PRINTING THE TEXT ARE UNPREDICTABLE. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE *m*th WITHTEXT OF THE COMMAND. PROCESSING CONTINUES.**

Insert Explanation:

command DRAWBOX or DRAWCIRCLE.
nn Line number.
*n*th Number of text line within WITHTEXT.
*m*th Number of WITHTEXT within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0540I **EXTENT - WARNING:**
THE SETTEXT **COMMAND BEGINNING ON LINE *nn* SPECIFIES A LINE OF TEXT THAT EXTENDS PAST THE *direction* OVERLAY BOUNDARY BY *value* PELS. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND. NO TEXT IN THIS LINE WAS PRINTED. PROCESSING CONTINUES.**

Insert Explanation:

nn Line number.
direction TOP, BOTTOM, LEFT or RIGHT.
value Number of pels in error.
nth Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0542I UNDERSCORE - WARNING:
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES FONT *name* FOR UNDERLINED TEXT. THIS FONT CANNOT BE USED FOR UNDERLINED TEXT. THE LINE IN ERROR IS THE *nth* TEXT LINE SPECIFIED IN THE COMMAND. THE TEXT IN THIS LINE USING FONT *name* WAS PRINTED BUT NOT UNDERLINED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
name Font name specified.
nth Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0543I FONT ORIENTATION - WARNING:
THE SETTEXT COMMAND BEGINNING ON LINE *nn* PECIFIES FONT *name* FOR TEXT IN THE *format* FORMAT, ORIENTED AT *degrees* DEGREES. FONT *name* CANNOT BE USED FOR TEXT IN THAT COMBINATION OF FORMAT AND ORIENTATION. THE LINE IN ERROR IS THE *nth* TEXT LINE SPECIFIED IN THE COMMAND. THE TEXT IN THIS LINE USING FONT *name* WAS NOT PRINTED. THE SYSTEM PROGRAMMER CAN HELP YOU DETERMINE THE CORRECT FONT TO USE.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
name Font name specified.
format MODERN, COLUMN or TATE.
degrees Text orientation.
nth Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0544I UNDERSCORE - WARNING:
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES THAT TEXT IN THE *format* FORMAT IS TO BE UNDERLINED. TEXT IN EITHER THE COLUMN OR TATE FORMAT CANNOT BE UNDERLINED. THE LINE IN ERROR IS THE *nth* TEXT LINE SPECIFIED IN THE COMMAND. THE TEXT IN THIS LINE WAS PRINTED BUT NOT UNDERLINED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
format COLUMN or TATE.
n Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0545I **DOUBLE-BYTE FONT - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES AN ODD NUMBER OF BYTES TO BE PRINTED WITH FONT *name* WHICH IS A DOUBLE-BYTE FONT. THE TEXT IN ERROR IS SPECIFIED IN THE *n*th TEXT LINE OF THE COMMAND. THE RESULTS OF TRANSLATING AND PRINTING THE TEXT ARE UNPREDICTABLE.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
name Font name specified.
*n*th Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0546I **TOO MANY FONTS - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIES FONT *name* FOR TEXT IN THE *format* FORMAT, ORIENTED AT *degrees* DEGREES. YOU ARE OVER THE LIMIT OF *value* UNIQUE COMBINATIONS OF FONT, TEXT FORMAT AND TEXT ORIENTATION THAT MAY BE USED IN A SINGLE OVERLAY. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND. THE TEXT IN THIS LINE USING FONT *name* WAS NOT PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
name Font name specified.
format MODERN, COLUMN, or TATE.
degrees Text orientation.
value Maximum number of font combinations.
*n*th Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0548I **HEXADECIMAL - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIED A TEXT STRING WITH THE HEX KEYWORD. THE TEXT STRING EITHER CONTAINED AN ODD NUMBER OF CHARACTERS OR CONTAINED A CHARACTER THAT WAS NOT HEXADECIMAL. THE RESULTS OF TRANSLATING AND PRINTING THE TEXT ARE UNPREDICTABLE. THE LINE IN ERROR IS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th Number of text line within command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0559I **NOTIFY - ERROR:**
THE FONT COMMAND BEGINNING ON LINE *nn* SPECIFIES A FONT FOR WHICH NO INFORMATION WAS FOUND FOR ANY SUPPORTED ORIENTATION. THE FONT NAME SPECIFIED WAS "*member*"; THE FONT TYPE USED WAS "*type*". ENSURE THAT THE FONT NAME WAS SPECIFIED CORRECTLY.
FOR ADDITIONAL HELP, NOTIFY THE SYSTEM PROGRAMMER.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
member Member ID specified.
type Member type specified.

System Action: Processing continues.

User Response: Notify the system programmer.

DZI0560I **BLANK PAGE - ERROR:**
THE OVERLAY, AS SPECIFIED, CONTAINS NO VALID *type_command* COMMAND; THEREFORE,
NO OVERLAY WAS GENERATED.
PROCESSING CONTINUES.

Insert Explanation:

type_command OVERLAY or ACTION

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0561I **FONT - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIED A TEXT STRING CONTAINING
***text_type* CHARACTERS. NO *text_type* FONT WAS SPECIFIED FOR THIS LINE. THE LINE IN**
ERROR WAS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND. THE TEXT WAS NOT PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
text_type SBCS or DBCS.
*n*th Number of text line within SETTEXT command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0562I **FONT - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIED A TEXT STRING CONTAINING AN
SO OR SI CHARACTER WITH A SPECIFIED MODE OF SOSI1. NO SBCS FONT WAS SPECIFIED
FOR THIS LINE. THE SO OR SI CHARACTER(S) WERE NOT REPLACED WITH A BLANK SPACE.
THE LINE IN ERROR WAS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND.
PROCESSING CONTINUES.

Explanation: Mode SOSI1 specifies that SOSI characters are to be replaced by a blank space using the SBCS font.

Insert Explanation:

nn Line number.
*n*th Number of text line within the SETTEXT command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0563I **FONT - WARNING:**
THE SETTEXT COMMAND BEGINNING ON LINE *nn* SPECIFIED *font1* AND *font2* AS A FONT PAIR.
THESE ARE BOTH *font_type* FONTS. A FONT PAIR MUST BE ONE OF EACH TYPE. ONLY *font1* IS
ACCEPTED. THE LINE IN ERROR WAS THE *n*th TEXT LINE SPECIFIED IN THE COMMAND.
PROCESSING CONTINUES.
Note: If a font of the type NOT specified in this pair was specified previously in this text line, it is
still in effect.

Insert Explanation:

nn Line number.
font1 First font specified in pair.
font2 Second font specified in pair.
font_type SBCS or DBCS.
*n*th Number of text line within the SETTEXT command.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0580I **DRAWPATH - WARNING:**
THE DRAWPATH COMMAND BEGINNING ON LINE :mv.nn:emv. SPECIFIED A "TO"
SUBCOMMAND THAT DEFINED THE SAME LOCATION AS THE PREVIOUS "TO" SUBCOMMAND.
THE REDUNDANT SPECIFICATION IS IGNORED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0582I **DRAWPATH - WARNING:**
THE DRAWPATH COMMAND BEGINNING ON LINE *nn* SPECIFIED A ROUNDED CONNECTION FOR
THE *n*th "TO" SUBCOMMAND. THE CORNER LENGTH FOR THIS CONNECTION, *lenvalue*, IS
LARGER THAN THE MAXIMUM POSSIBLE FOR THIS CONNECTION, *maxvalue*. *maxvalue* WAS
USED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

*n*th Number of TO subcommand within this command.

lenvalue Corner length defined in CONNECTION subcommand or SETUNITS command.

maxvalue Largest corner length possible for this connection.

System Action: Processing continues.

User Response: Refer to the section of the User Guide and Reference where MAX for connections is explained.

DZI0584I **DRAWPATH - WARNING:**
THE DRAWPATH COMMAND BEGINNING ON LINE *nn* SPECIFIED SHADING FOR A PATH THAT IS
OPEN. SHADING CAN ONLY BE DONE ON A CLOSED PATH.
PROCESSING CONTINUES.

Explanation: Shading can only be specified for a closed path. That is, the start and end points of the path must be the same and there must be no breaks in the path because of large rounded connections.

Insert Explanation:

nn Line number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0586I **EXTENT - WARNING:**
IN THE DRAWPATH COMMAND BEGINNING ON LINE *nn*, THE *n*th PATH EXTENDS PAST THE
***direction* OVERLAY BOUNDARY BY *value unit* . THE PORTION OF THE PATH OUTSIDE THE**
OVERLAY WILL NOT BE PRINTED.
PROCESSING CONTINUES.

Insert Explanation:

nn Line number.

*n*th Path number.

direction LEFT, TOP, RIGHT or BOTTOM.

value Amount of error.

unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0587I **EXTENT - WARNING:**
IN THE DRAWPATH COMMAND BEGINNING ON LINE *nn*, THE *n*th PATH, AS SPECIFIED, WOULD BE ENTIRELY OUTSIDE THE *direction* OVERLAY BOUNDARY BY *value unit*, MEASURED FROM THE *direction*-MOST POINT OF THE PATH. THIS PATH WILL NOT BE PRINTED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th Path number.
direction TOP, LEFT, RIGHT or BOTTOM.
value Amount of error.
unit Unit of measurement.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0588I **DRAWPATH - WARNING:**
IN THE DRAWPATH COMMAND BEGINNING ON LINE *nn*, SHADING WAS SPECIFIED FOR THE *n*th PATH. THAT PATH EXTENDS PAST THE OVERLAY BOUNDARY. THE PATH WILL NOT BE SHADED. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
*n*th Path number.

System Action: Processing continues.

User Response: Correct the error and resubmit the job.

DZI0589I **DRAWPATH - INFORMATIONAL:**
THE DRAWPATH COMMAND BEGINNING ON LINE *nn* SPECIFIED A PATH WITH WIDTH *width units*, MEASURED FROM THE LEFT_MOST TO THE RIGHT-MOST POINTS OF THE PATH. THE HEIGHT OF THE PATH IS *height units*, MEASURED FROM THE TOP-MOST TO THE BOTTOM-MOST POINTS OF THE PATH. PROCESSING CONTINUES.

Insert Explanation:

nn Line number.
width Width of the path.
height Height of the path.
units Unit of measurement.

System Action: Processing continues.

User Response: None.

DZI0590I **FONT - WARNING:**
THE FONT COMMAND BEGINNING ON LINE *nn*SPECIFIES BOTH A CODED FONT NAME AND A CHARSET/CODEPAGE COMBINATION. ONLY ONE OPTION IS ALLOWED. THE CHARSET/CODEPAGE COMBINATION WILL BE USED. PROCESSING CONTINUES

Insert Explanation:

nn Line number

System Action: Processing continues using the character set/codepage combination.

User Response: Remove either the coded font name or the character set/codepage parameter and resubmit the job.

DZI0592I **FONT - ERROR:**
THE FONT COMMAND BEGINNING ON LINE *nn* USES THE CHARSET/CODEPAGE OPTION WITH AN UNBOUNDED BOX FORMATTED FONT. THIS IS NOT SUPPORTED. CHANGE THE COMMAND TO USE THE CODED FONT NAME AND RESUBMIT THE JOB. THIS COMMAND IS IGNORED. PROCESSING CONTINUES

Insert Explanation:
nn Line number

System Action: Processing continues without this font command.

User Response: Change the character set/codepage parameters to point to a bounded box font or use the coded font parameter to specify the needed font and resubmit the job.

DZI0601I **CBDUMP - INFORMATIONAL MESSAGE:**
THE OVERLAY CANNOT BE STORED OR REPLACED WHEN THE CBDUMP COMMAND HAS BEEN ISSUED. PROCESSING CONTINUES.

System Action: Processing continues.

User Response: None.

DZI0703I **FINAL DISPOSITION:**
THE OVERLAY *action1* IN THE OVERLAY LIBRARY. THE SAMPLE OVERLAY *action2* AVAILABLE FOR PRINTING. OPTIMIZATION OF IMAGE DATA *action3* PERFORMED.
MESSAGE SEVERITY SUMMARY:
count1a INFORMATIONAL MESSAGES WERE SUPPRESSED.
count1b INFORMATIONAL MESSAGES WERE PRINTED.
count2a WARNING MESSAGES WERE SUPPRESSED.
count2b WARNING MESSAGES WERE PRINTED.
count3 ERROR MESSAGES WERE PRINTED.
FINAL RETURN CODE WAS: *rc*

Insert Explanation:
action1 WAS REPLACED or WAS STORED or WAS NOT STORED.
action2 IS or IS NOT.
action3 WAS or WAS NOT.
count1a Number of informational messages suppressed by the utility.
count1b Number of informational messages printed by the utility.
count2a Number of warning messages suppressed by the utility.
count2b Number of warning messages printed by the utility.
count3 Number of error messages printed by the utility.
rc Final return code.

System Action: Processing continues.

User Response: Correct errors, if necessary, and resubmit the job.

DZI0708I **FINAL DISPOSITION:**
OVERLAY FILE: *action1*
SAMPLE OVERLAY FILE: *action2*
IMAGE OPTIMIZATION: *action3*
MESSAGE SEVERITY SUMMARY
count1a INFORMATIONAL MESSAGES WERE SUPPRESSED.
count1b INFORMATIONAL MESSAGES WERE PRINTED.
count2a WARNING MESSAGES WERE SUPPRESSED.
count2b WARNING MESSAGES WERE PRINTED.
count3 ERROR MESSAGES WERE PRINTED.
FINAL RETURN CODE: *rc*

Insert Explanation:
action1 NOT CREATED or CREATED or CREATED AND PRINTED or REPLACED.

action2 NOT CREATED or CREATED or CREATED AND PRINTED or REPLACED.
action3 PERFORMED or NOT PERFORMED.
count1a Number of informational messages suppressed by the utility.
count1b Number of informational messages printed by the utility.
count2a Number of warning messages suppressed by the utility.
count2b Number of warning messages printed by the utility.
count3 Number of error messages printed by the utility.
rc Final return code.

System Action: Processing continues.

User Response: Correct errors, if necessary, and resubmit the job.

**DZI0996I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE MESSAGE *msgid* IS MISSING FROM THE MESSAGE CSECT.
PROCESSING CONTINUES.**

Insert Explanation:

msgid Message identifier.

System Action: Message identifier.

User Response: Notify the system programmer.

**DZI0997I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE ENTRY FOR THE KEYWORD WITH KEYWORD CODE *value* IS MISSING FROM THE
KEYWORD CSECT. THE KEYWORD IS REPLACED BY ASTERISKS IN THE MESSAGE BEING
BUILT.
PROCESSING CONTINUES.**

Insert Explanation:

value Keyword number.

System Action: Processing continues.

User Response: Notify the system programmer.

**DZI0998I NOTIFY - ERROR:
NOTIFY THE SYSTEM PROGRAMMER.
THE ENTRY FOR THE INSERT WITH INSERT CODE *value* IS MISSING FROM THE *csect* CSECT.
THE INSERT IS REPLACED BY ASTERISKS IN THE MESSAGE BEING BUILT.
PROCESSING CONTINUES.**

Insert Explanation:

value Insert number.

csect INSERT or MESSAGE.

System Action: Processing continues.

User Response: Notify the system programmer.

Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property rights may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Printing Systems Division
Department H7FE Building 003G
Information Development
PO Box 1900
Boulder CO 80301-9191 USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee. The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

For online versions of this book, we authorize you to:

- Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy.
- Transfer the original unaltered copy of the documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for payment of any taxes, including personal property taxes, resulting from this authorization.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine readable documentation.

Programming Interfaces

This publication includes documentation of intended Programming Interfaces that let the customer write programs to obtain the services of the Page Printer Formatting Aid (PPFA).

Trademarks

The following terms appear in this publication and are either trademarks or registered trademarks of the IBM Corporation:

IBM®
Advanced Function Presentation™
AFP™
AIX

Advanced Function Common Control Unit
Advanced Function Presentation
Advanced Function Printing
AFCCU
AFP
AIX
AIX/6000
OS/400
Bar Code Object Content Architecture
BCOCA
IBM
Infoprint®
Infoprint Server for OS/390®
Intelligent Printer Data Stream
IPDS
Mixed Object Document Content Architecture
MO:DCA
PrintManager
Print Services Facility
PSF
OS/400
RISC System/6000
S/370

The following terms appear in this publication and are trademarks of other companies:

Netscape® Netscape is a registered trademark of Netscape Communications Corporation

Netscape Navigator® Netscape Navigator is a registered trademark of Netscape Communications Corporation

Windows® and Windows NT®

Windows and Windows NT are registered trademarks of the Microsoft Corporation.

Internet Explorer™

Internet Explorer is a trademark of Microsoft, Corp.

UNIX™

UNIX is a trademark of System Laboratories, licensed exclusively by X/Open Company, Ltd.

HP-UX

HP-UX is a trademark of Hewlett Packard Company.

Sun Solaris

Sun Solaris is a trademark of Sun Microsystems, Inc.

EuroReady

The Overlay Generation Language (OGL) is capable of processing data containing the euro sign. Font character sets and code pages that contain and map the euro sign consistently with the application must be present either in a host library or in the printer. AFP fonts that support the euro sign are included in the AFP Font Collection (Program Number 5648-B33).

Glossary

Glossary

This glossary defines terms, abbreviations, and acronyms as they are used in this manual. Also included are terms and definitions from:

- *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.
- *ISO Vocabulary of Office Machines*. Definitions from published sections of this vocabulary are identified by the symbol (ISO).
- *American National Dictionary for Information Processing*. These definitions are identified by an asterisk (*).
- *Dictionary of Computing*, SC20-1699.

A

absolute positioning. Positioning an element of an overlay with respect to the overlay origin. If the element is part of a defined group, it is positioned with respect to the group origin.

addressable point. In computer graphics, any point in a display space or tablet that can be addressed.

Advanced Function Presentation (AFP). The ability of program products to use the all-points-addressable concept, to print text and illustrations on a printer.

AFP. Advanced Function Presentation.

alphanumeric data. Data represented by letters and digits, perhaps with special characters and the space character. (I) (A)

ascender. (1) In a font, the distance from the baseline to the top of the character box. This value is the same for all the characters in a given font. See *maximum ascender*. (2) The part of a lowercase letter that rises above the body of the letter. Letters with ascenders are b, d, f, h, k, l, and t.

B

baseline. An imaginary line upon which successive characters are aligned.

bevel connection. A type of path connection that is drawn by the DRAWPATH command. A bevel connection is an angular connection which is automatically used in place of a miter connection when path segments join at a small angle.

boldface. A heavy-faced type. Such as, **bold**.

border. (1) The margins of a box. (2) The margin of a circle.

border thickness. The thickness of the border of a box, circle, or path.

border weight. Synonymous with *border thickness*.

C

character. A symbol used in printing. For example, a letter of the alphabet, a numeral, a punctuation or any other symbol that represents information.

character box. The boundary completely surrounding the character pattern.

character set. A set of unique representations called characters, for example the 26 letters of the English alphabet, 0 and 1 of the Boolean alphabet, the set of signals in the Morse code alphabet, and the 128 ASCII characters.

characters per inch (CPI). The number of characters of a given font that can fit into one linear inch.

closed paths. Paths created by the DRAWPATH command may be either *open* or *closed*. A closed path is one in which the last point in the path is the same as the first point in the path. The POSITION command preceding the DRAWPATH command specifies the start of the path. Only closed paths can be shaded. See also 398 and 396.

code page. A font component that associates code points and character identifiers. A code page also identifies how undefined code points are handled.

code point. An 8-bit binary number representing one of 256 potential characters.

coded font. (1) A font component that associates a code page and a font character set. (2) A font that is fully described in terms of typeface, point size, weight, width, and attribute.

coded overlay. An overlay sent to the printer as a command stream. Contrast with *raster pattern* overlay.

command line. On a terminal display, the line where commands are entered.

command stream. An overlay definition.

comment. Descriptions added to an overlay definition that describe what the commands are doing but do not affect the way the overlay is printed.

continuous forms paper. A continuous length of single-ply, fan-folded paper with both edges punched for

tractor feeding and with perforation between pages. There are various sizes and basis weights.

coordinate. The horizontal or vertical distance from an established point to the origin of an overlay element. The established point could be the overlay origin, a group origin, or the point specified in the last POSITION command.

corner arc. An arc of a circle formed by each rounded box corner or rounded path connection.

corner length. The degree of rounding of a corner. Each rounded box corner and rounded path connection drawn by OGL/370 is an arc of a circle. The corner length is the distance between the corner point and the point on the straight line where the corner arc begins. For a box and a right-angle path connection, the corner length is the same as the radius of the corner arc.

CPI. Characters per inch.

cut-sheet paper. Paper that is cut into sheets before being printed on. Contrast with 395.

D

data definition name (DDname). The name of a data definition statement.

data definition statement. A JCL statement that defines a data set. For example, a font library or a segment library.

data file. See 396.

data set. In MVS and VSE, the major unit of data storage and retrieval in the operating system. Consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DBCS. See 396.

DD statement. Data definition statement.

DDname. Data definition name.

default. Synonym for 396.

default value. The value assumed when none is specified.

definition. See 398.

descender. (1) In a font, the distance from the baseline to the bottom of the character box. This value might differ for different characters in a given font. However, OGL/370 uses the largest descender (see 398) in the font to determine if a text string fits in a box or overlay. (2) The part of a letter that falls below the body of the letter. Letters with descenders are g, j, p, q, n, y, and Q.

disjointed path. Paths created by the DRAWPATH command where the points define a *closed* path, but where the size of the rounded connections within the path have made the path open. A disjointed path cannot be shaded. See also 395 and 398 paths.

double-byte character set (DBCS). A character set, such as Kanji, requiring two bytes to identify each character.

double-byte coded font. A font in which the characters are defined by two bytes: the first defining a coded font section, the second defining a code point.

double-byte font. Synonym for 396.

duplex. Pertaining to printing on both sides of a sheet of paper. See 398 and 401. Contrast with 400.

E

electronic overlay. An overlay that is stored in a library and can be requested for a printing job. See also 397, 399.

encoded. A way of representing an image pattern so that toned and untoned pels are grouped together rather than each being represented in the overlay definition by a 1 or 0.

end marker. The semi-colon (;) that must appear at the end of each command.

entry. A keyword, name, or value that is part of a command.

error message. A message indicating that the overlay definition contains an error which may cause a command to be ignored or the overlay not to be printed.

F

figure. A shape which is part of an overlay. Circles, boxes, paths, and rules are figures but pieces of text are not.

file. A set of related data records.

fixed-space font. Synonym for 401..

FLSF. Font Library Service Facility

font. (1) Refers to one size and one typeface in a particular type family. Includes letters, numbers, punctuation marks, special characters, and ligatures. (2) A font component that identifies a character set, code page combination. (3) See 401.

font character set. (1) A font library member that contains the raster patterns, identifiers, and descriptions of characters. (2) Synonymous with 395.

font definition. A listing of the printing attributes of a coded font. The font definition is an internal library member, built into an external library object called a coded font.

form. An overlay.

format. The arrangement of text on the page.

form definition. A resource object that defines the characteristics of the form which include: overlays to be used (if any), text suppression, the position of page data on the form, and the number and modifications of a page. Synonymous with 397.

FORMDEF. Synonym for 396.

form map. Synonym for 396.

forms flash. On the IBM 3800, a means of printing an overlay using a negative plate flashed onto the form. Synonymous with optical forms flash.

forms overlay. The photographic negative of a predefined design to be exposed to the photoconductor by a flash of light. The forms overlay can be merged with variable data during printing. Synonymous with 398.. See also 396.

frame. A border around an overlay.

G

GDDM. Graphical Data Display Manager.

GOCA. Graphical Object Content Architecture

graphic. Image or text, or a combination of both that can be placed on an overlay by name.

Graphical Data Display Manager (GDDM). An IBM program product that creates page segments.

grid. Synonym for 398.

group. A named collection of commands, which can be placed in an overlay by name.

H

Hiragana. A character set consisting of symbols used in one of the two common Japanese phonetic alphabets. Each character is represented by one byte.

hexadecimal. A representation of the way data is read by the computer. Specifically in OGL/370, the code used to represent text.

I

ID. The last six characters of a member name.

ID name. The internal name given to a font by a user.

image. A pattern of toned and untoned pels that form a picture.

informational message. A message saying that a command is not entirely correct or complete. The appearance of the overlay probably is not affected.

invocation. The statement which activates the OGL/370 program in the VM environment.

J

JCL. Job control language.

JCS. Job control statements.

job control language (JCL). A language of control statements used to identify a computer job or describe its requirements to the operating system.

job control statement (JCS). A statement that provides an operating system with information about the job being run under VSE.

justify. (*) To align characters horizontally or vertically to fit the positioning constraints of a required format.

K

Kanji. The non-phonetic Japanese writing system. In a font representing Kanji characters, each character is represented by a double-byte code. Contrast with 397 and 397.

Katakana. A character set consisting of symbols used in one of the two common Japanese phonetic alphabets. Each character is represented by one byte.

Kerning. The design of graphic characters such that their character boxes overlap. Some toned pels of the character appear outside the character cell. The reduction of space between letters.

keyword. A word in OGL/370 that must be entered exactly as shown and may not be used as a name for a font, segment, definition, or overlay. Also referred to as a command word or subcommand word.

L

librarian. In VSE, the set of programs which maintains, services, and organizes the system and private libraries.

library. (1) A named area on disk that contains a collection of related files, of which the computer can access. (2) A place to store resource objects.

line. A straight geometrical element.

lines per inch (LPI). (1) A unit of measurement for the specification of baseline placement. (2) A measure of the number of lines per vertical inch of paper.

line printer. (1) (ISO) A device that prints a line of characters as a unit. (2) Contrast with 399.

line space. The vertical distance between the baseline of the current line and the baseline of the previous line.

location repetition. Repetition of lines or boxes within the same command when the position of each line or box is specified. Contrast with 400.

logical page. The area defined as the space on the :hp1.physical page:ehp1. where data is to be printed.

LPI. Lines per inch.

M

mask. Horizontal and vertical lines printed on an overlay to help in the design of the overlay. Synonymous with 397.

maximum ascender. The maximum height from the baseline to the top mark of any character in the font character set.

maximum descender. The maximum depth from the baseline to the bottom mark of any character in the character set.

member. A file in a library. For example, font X1BITR is a member of the font library.

member ID. The member name of a resource object minus the two-character prefix. For example BITR is the member ID of the font whose member name is X1BITR.

member name. The name under which a file is stored in a library. For example X1S0BITR is the member name of a coded font in the font library.

mirror image. Converting page segments so that their component parts appear reversed when compared to the original.

miter connection. A type of path connection used in the DRAWPATH command. Miter connections between path segments are angular. Contrast with rounded connection.

mm. Millimeter.

modern. The term used to describe left-to-right, top-to-bottom writing or printing for Kanji fonts.

multiple virtual storage (MVS). An IBM operating system.

MVS. Multiple Virtual Storage.

N

negative image. Converting segments, so what appeared black in the original, appears white and what appeared white in the original, appears black.

normal duplex. Printing on both sides of the paper so that the top of one side is at the same end as the top of the other. Contrast with 400, 401.

O

offset. The coordinates of the corner of the overlay nearest to the paper origin.

open paths. Paths created by the DRAWPATH command may be either *open* or *closed*. An open path is one in which the last point in the path is not the same as the first point in the path. The POSITION command preceding the DRAWPATH command specifies the start of the path. Open paths cannot be shaded. See also 395 and 396 paths.

OGL/370. Overlay Generation Language/370.

optical forms flash. Synonym for 397.

optical forms overlay. Synonym for 397.

option. A word or group of words that can be used to specify a modification to an overlay element. For example, the shading option in the DRAWBOX command.

orientation. The rotation of an element relative to a fixed reference.

origin. (1) The point in an element that is used for positioning the element. (2) The upper-left corner of an overlay in its 0° orientation. (3) A position from which placement and orientation of text, images, and page segments is specified. Sheets of paper, pages, overlays, and page segments have origins.

overlay. A collection of predefined data, such as lines, shading, text, boxes, or logos, that can be merged with variable data on a sheet while printing. An overlay can be either 396 or 397.

overlay definition. The collection of commands that define an overlay.

Overlay Generation Language/370 (OGL/370). An IBM program product used to create electronic overlays.

overlay section. (OGL/370) A section of an overlay that can be placed by name (a group, segment, or pattern).

P

page. Synonym for 398.

PAGEDEF. Page definition.

page definition (PAGEDEF). A resource, specified in the print data set JCL, that defines the rules for transforming the input to pages and text controls.

page printer. (1) Any printer that accepts composed pages, constructed of composed text and images, among other things. (2) Contrast with 398.

Page Printer Formatting Aid (PPFA). An IBM program product that allows for creation and storage of form definitions and page definitions—resource objects for print-job management.

page segment. An object containing composed text and images, prepared before formatting and included during printing. Synonymous with 400. Contrast with 396.

paper origin. The upper-left corner of the paper (excluding the 1/2-inch carrier strip on the left margin) as the paper goes through the printer.

parse. To analyze the operands entered with a command and build a parameter list for the command processor.

path. One or more connected straight lines of a given thickness and style.

pattern. See 399.

pel. The smallest area that can be individually toned by the printer. On the IBM 3800 Model 1, one pel equals approximately 1/180th inch. On the IBM 3800 Model 3 and Model 8, one pel equals approximately 1/240th inch. On the IBM 4250, one pel equals approximately 1/600th inch. Synonymous with 395.

physical page. The actual sheet of paper or other medium (such as a sheet of blank labels) that moves through the printer.

picture element. Synonym for 399.

pitch. A unit of measurement for the width of a printed character, reflecting the number of times a graphic character can be set in one linear inch; for example, 10-pitch has 10 graphic characters per inch. Uniformly spaced fonts are measured in pitches. Contrast with point.

PMF. Print Management Facility.

point. (1) A unit of measurement. There are 12 points to a pica and approximately 72 points to an inch. (2) In the Didot point system, a point is 0.0148 inches. There are 12 Didot points to the cicero. (3) Contrast with 399. (4) See also 401.

point size. The height of a font in points.

PPFA. Page Printer Formatting Aid.

preprinted form. A sheet of paper containing a preprinted design of constant data into which variable data can be merged. See also 396 and 397.

primary default. The first default specified in the SETUNITS command.

print line. In a display image, a horizontal line at the top or bottom of a string of characters.

Print Management Facility (PMF). A program that can create fonts, segments, page definitions, and form definitions.

print position. A pel identified by a set of coordinates.

Print Services Access Facility (PSAF). A menu-driven, print-parameter selection program for page printers controlled by PSF.

Print Services Facility (PSF). A program that produces printer commands from the data sent to it.

printable area. The area of the paper where print can be placed.

proportional spacing. The spacing of characters in a printed line so that each character is allotted a space based on the character's width.

proportionally spaced font. (1) A font in which the characters are contained in character cells that vary with the size of each character. This allows for even spacing between printed characters, and eliminates excess white space around narrow characters, such as the letter "i". (2) Contrast with 396.

PSAF. Print Services Access Facility.

PSF. Print Services Facility.

R

raster pattern. A series of pels arranged in scan lines. The toned or untoned status of each pel creates an image. A digitized raster pattern is an array of bits. The on or off status of each bit determines the toned or untoned status of each pel.

raster pattern overlay. An overlay loaded in the printer subsystem as a raster pattern rather than as a sequence of printer commands.

record. A collection of related data or words, treated as a unit. For example, in stock control, each invoice could constitute one record.

relative positioning. Positioning an element of an overlay with respect to the last position established by the last POSITION command.

rotated font. A font whose characters are rotated 90 or 270 degrees to allow for printing at those orientations.

rotation. The number of degrees a character is rotated relative to the print direction.

rounded connection. (1) A type of path connection used in the DRAWPATH command. Rounded connections between path segments are curved according to a corner length value specified in the SETUNITS command or the CONNECTION subcommand. (2) Contrast with 398.

rule. A straight horizontal or vertical line.

S

SBCS. Single-byte character set.

screen. The OGL/370 shade pattern option containing a consistent combination of toned and untoned pels.

secondary default. The second default specified in the SETUNITS command.

segment. Synonym for page segment.

shade. The option available with DRAWBOX, DRAWCIRCLE, DRAWPATH, and PLACE PATTERN to tint part of the overlay with a selected intensity of gray.

shift-in character (SI). Used to identify the end of DBCS characters in a mixed (DBCS/SBCS) text string. See 400.

shift-out character (SO). Used to identify the start of DBCS characters in a mixed (DBCS/SBCS) text string. See 400.

SI. Shift-in character.

simplex. Printing on only one side of the paper. Contrast with 396 and 398.

single-byte character set (SBCS). A character set whose codes require a single byte of data. For example, the character set used in English.

single-byte coded font. (1) A font in which the characters are defined by a one byte code point. A single byte coded font has only one coded font section. (2) Contrast with 396.

single-byte font. Synonym for 400.

SO. Shift Out character.

source listing. A listing of the overlay definition and messages after OGL/370 has processed the definition.

spaced repetition. (1) Repetition of lines or boxes within the same command when the set spacing between each line or box is specified. (2) Contrast with 398.

standard shade pattern. The default shade pattern used when shading with DRAWBOX, DRAWCIRCLE, DRAWPATH, and PLACE PATTERN.

string. (1) A linear sequence of entities such as characters or physical elements. (2) See 400.

subcommand. A keyword that introduces a distinct part of a command. For example, the REPEAT subcommand of the DRAWBOX and DRAWRULE commands.

symbolic. A name that identifies a text string that can be called for by name in DRAWBOX WITHTEXT, DRAWCIRCLE WITHTEXT, or SETTEXT.

symbolic data set. In MVS and VSE, a data set containing text strings associated with symbolic names. These strings can be called for by name in DRAWBOX WITHTEXT, DRAWCIRCLE WITHTEXT, or SETTEXT.

symbolic file. In VM, a data file containing text substitutions for symbolics used in DRAWBOX WITHTEXT, DRAWCIRCLE WITHTEXT, and SETTEXT commands.

syntax. The rules and keywords of OGL/370.

system programmer. A programmer who plans, generates, maintains, and controls the use of the OGL/370 program product at your installation.

T

tate. Pronounced tah-tay. The traditional Japanese format of top-to-bottom, right-to-left printing for Kanji text.

text block. All the lines of text specified in one WITHTEXT subcommand or SETTEXT command.

text margin. The boundary of the area where text can be placed inside a box or a circle. A rounded text margin is one border width inside the border for both circles and boxes. A square text margin is the largest rectangle that fits inside the rounded margin.

text segment. One section of a *text string* enclosed in apostrophes.

text string. (1) The *text segment* or segments, entered within one LINE subcommand. Each time a LINE subcommand is entered, a new *text string* begins. (2)

Text that is to be printed as part of the overlay. It can appear in DRAWBOX WITHTEXT, DRAWCIRCLE WITHTEXT, and SETTEXT. Each *text segment* of the *text string* is enclosed in apostrophes.

tumble duplex. (1) Duplex printing for sheets that are to be bound on the short edge of the paper, regardless of whether the printing is portrait or landscape. (2) Contrast with 398 and 400.

typeface. A collection of fonts all having the same style, weight, and width; each font differs from the others by point size.

type family. (Fonts) A group of typefaces that share basic design characteristics and encompass many size and style variations.

type font. Type of a given size and style. For example, 10-point Bodoni Modern.

typographic font. A family of proportionally spaced characters that produce output resembling typeset copy rather than typewritten copy.

U

uniformly spaced font. (1) A font with graphic characters contained in character cells of uniform size. The distance between reference points of adjacent graphic characters is constant in the inline progression. The white space between the graphic characters may vary. (2) Synonymous with 396. (3) Contrast with 399.

unprintable area. The area of a sheet of paper on which no printing can be done because of printer-hardware limitations.

V

value. A number or name that a user specifies in a command.

variable space font. See 399.

virtual machine (VM). A functional simulation of a computer and its associated devices.

Virtual Storage Extended (VSE). An operating system that is an extension of DOS/VS, consisting of VSE/Advanced Functions.

Virtual Storage Extended/Advanced Functions (VSE/AF). The minimum operating system support for a VSE-controlled installation.

VM. Virtual Machine.

VSE. Virtual Storage Extended

VSE/AF. Virtual Storage Extended/Advanced Functions (VSE/AF)

W

warning message. A message saying the overlay definition contains an error that will probably result in undesirable output.

X

X axis. The axis that is perpendicular to the direction the paper moves through the printer.

x-coordinate. The horizontal or inline position that defines a page origin or the starting point of a line or field.

Y

Y axis. The axis along which the paper moves through the printer.

y-coordinate. The vertical or baseline position that defines a page origin or the starting point of a line or field.

Index

Special characters

- absolute positioning* subcommand
 - for POSITION command 280
- alignment* subcommand
 - for SETTEXT command 287
- axis selection* subcommand
 - for DRAWBOX command
 - text* 220
- border thickness* subcommand
 - for DRAWBOX command 208, 219
- border type* subcommand
 - for DRAWBOX command 208, 219
- box height* subcommand
 - for DRAWBOX command 208
- box width* subcommand
 - for DRAWBOX command 207
- circle portion* subcommand
 - for DRAWBOX command 219
- circle radius* subcommand
 - for DRAWBOX command 218
- colorname* subcommand
 - for DEFINE COLOR command 199
 - for DRAWBOX command 209
 - for DRAWGRAPHIC command 238
- colorname, SHADE* subcommand
 - for DRAWBOX command 212
- command word* subcommand
 - for DEFINE GROUP command 201
- connection length* subcommand
 - for DRAWPATH command 245, 248
- connection type* subcommand
 - for DRAWPATH command 245, 247
- corner selection* subcommand
 - for DRAWBOX command 209
- depth*
 - OBJECT subcommand
 - for PLACE command 276
- diagonal selection* subcommand
 - for DRAWBOX command 209, 220
- direction* subcommand
 - for DRAWBOX command
 - spaced repetition 210
 - for DRAWCIRCLE command
 - spaced repetition 221
 - for DRAWPATH command 249
 - for DRAWRULE command
 - spaced repetition 256
- fillpattern* subcommand
 - for DRAWGRAPHIC command 238
- first coordinate* subcommand
 - for DRAWPATH command 247
 - for POSITION command 280
- first spacing* subcommand
 - for DRAWMASK command 242
- font name* subcommand
 - for FONT command (MVS) 258
 - for FONT command (VM) 260
- font name* subcommand (*continued*)
 - for FONT command (VSE)) 262
- font names* subcommand
 - for SETTEXT command 289
- format* subcommand
 - for SETTEXT command 286
- group commands* subcommand
 - for DEFINE GROUP command 201
- horizontal coordinate* subcommand
 - for DRAWBOX command
 - location repetition 211
 - for DRAWCIRCLE command
 - location repetition 223
 - for DRAWPATH command
 - location repetition 250
 - for DRAWRULE command
 - location repetition 257
 - for OVERLAY command 268
- length value* subcommand
 - for SETUNITS command 293
- line coding* subcommand
 - for DEFINE OBJECT command 204
- line spacing* subcommand
 - for SETTEXT command 288
- linespace option* subcommand
 - for SETUNITS command 292
- linespace value* subcommand
 - for SETUNITS command 292
- linetype* subcommand
 - for DRAWGRAPHIC command 238
- linewidth* subcommand
 - for DRAWGRAPHIC command 238
- location option* subcommand
 - for DRAWCIRCLE command
 - location repetition 223
 - for DRAWRULE command
 - location repetition 257
- marker symbol* subcommand
 - for DRAWGRAPHIC command 238
- member ID* subcommand
 - for FONT command (MVS) 258
 - for FONT command (VM) 260
 - for FONT command (VSE)) 262
 - for SEGMENT command (MVS) 283
 - for SEGMENT command (VM) 284
 - for SEGMENT command (VSE) 285
- mirror option*
 - PATTERN subcommand
 - for PLACE command 279
- modifier* subcommand
 - for DEFINE BARCODE command 198
- name*
 - OBJECT subcommand
 - for PLACE command 276
- name* subcommand
 - for DEFINE BARCODE command 197
 - for DEFINE OBJECT command 202
 - for SEGMENT command (MVS) 283

name subcommand (*continued*)
 for SEGMENT command (VM) 284
 for SEGMENT command (VSE) 285
name, BARCODE subcommand
 for PLACE command 269
negative option
 PATTERN subcommand
 for PLACE command 279
orientation
 OBJECT subcommand
 for PLACE command 276
 PATTERN subcommand
 for PLACE command 278
orientation subcommand
 for FONT command (VSE)) 264
 for SETTEXT command 286
orientation, BARCODE subcommand
 for PLACE command 269
origin option subcommand
 for DRAWPATH command 247
 for POSITION command 280
overlay height subcommand
 for OVERLAY command 267
overlay name subcommand
 for OVERLAY command 267
overlay width subcommand
 for OVERLAY command 267
path end shape subcommand
 for DRAWPATH command 245
path start shape subcommand
 for DRAWPATH command 244
path thickness subcommand
 for DRAWPATH command 244
path type subcommand
 for DRAWPATH command 244
pattern type subcommand
 for DEFINE OBJECT command 204
positioning option subcommand
 for SETUNITS command 293
primary default subcommand
 for SETUNITS command 291
relative positioning subcommand
 for POSITION command 280
repetitions subcommand
 for DRAWBOX command
 spaced repetition 210
 for DRAWCIRCLE command
 spaced repetition 221
 for DRAWPATH command 249
 for DRAWRULE command
 spaced repetition 256
rule direction subcommand
 for DRAWRULE command 255
rule length subcommand
 for DRAWRULE command 255
rule thickness subcommand
 for DRAWRULE command 255
rule type subcommand
 for DRAWRULE command 255
second coordinate subcommand
 for DRAWPATH command 247
 for POSITION command 280
second spacing subcommand
 for DRAWMASK command 242
secondary default subcommand
 for SETUNITS command 292
section name
 PATTERN subcommand
 for PLACE command 278
section name subcommand
 for DEFINE GROUP command 201
 for DEFINE OBJECT command 204
 for PLACE command 275
section type
 for PLACE command 278
section type subcommand
 for DEFINE GROUP command 201
 for DEFINE OBJECT command 204
 for PLACE command 275
shade area subcommand
 for DRAWBOX command 212
 for DRAWCIRCLE command
 location repetition 224
shade pattern
 PATTERN subcommand
 for PLACE command 278
shade pattern subcommand
 for DRAWBOX command 212
 for DRAWCIRCLE command
 location repetition 224
shade type
 PATTERN subcommand
 for PLACE command 278
shade type subcommand
 for DRAWBOX command 212
 for DRAWCIRCLE command
 location repetition 224
spacing value subcommand
 for DRAWBOX command
 spaced repetition 210
 for DRAWCIRCLE command
 spaced repetition 221
 for DRAWPATH command 249
 for DRAWRULE command
 spaced repetition 256
spacing word subcommand
 for DRAWBOX command
 spaced repetition 210
text type subcommand
 for SETTEXT command 290
text type, BARCODE subcommand
 for PLACE command 269
text subcommand
 for SETTEXT command 290
type subcommand
 for DEFINE BARCODE command 197
vertical coordinate subcommand
 for DRAWBOX command
 location repetition 211
 for DRAWCIRCLE command
 location repetition 223

vertical coordinate subcommand (*continued*)
 for DRAWPATH command
 location repetition 250
 for DRAWRULE command
 location repetition 257
 for OVERLAY command 268
width
 OBJECT subcommand
 for PLACE command 276
x offset
 OBJECT subcommand
 for PLACE command 276
y offset
 OBJECT subcommand
 for PLACE command 276

Numerics

2DPARMS, BARCODE subcommand
 for PLACE command 269
 90° orientation 30

A

abbreviations 51
 for a keyword 338
 for units of measurement 191
 absolute POSITION 37
 absolute positioning, definition of 395
 adding text
 to boxes 100
 to circles 106
 with SETTEXT 108
 with symbolics 114
 addressable point, definition of 395
 AFP, definition of 395
 ALL messages 6
 ALL subcommand
 for CONTROL command 195
 alphanumeric data, definition of 395
 arc, corner 141
 ascender 99
 ascender, definition of 395
 ASTERISK, BARCODE subcommand
 for PLACE command 269
 AUTO
 line spacing 111
 AUTO line spacing
 in DRAWCIRCLE 98
 AXIS subcommand
 for DRAWBOX command 220

B

BALANCE 86
 balanced text
 DRAWBOX 86
 BARCODE
 DATAMATRIX, placing 270
 MAXICODE, placing 271
 PDF417, placing 272

BARCODE (*continued*)
 two-dimensional 270, 271, 272
 BARCODE *name* subcommand
 for PLACE command 269
 BARCODE *orientation* subcommand
 for PLACE command 269
 BARCODE *text type* subcommand
 for PLACE command 269
 BARCODE 2DPARMS subcommand
 for PLACE command 269
 BARCODE ASTERISK subcommand
 for PLACE command 269
 BARCODE DATAMATRIX subcommand
 special functions 271
 BARCODE HRI position subcommand
 for PLACE command 269
 BARCODE subcommand
 for PLACE command 269
 baseline 14
 baseline, definition of 395
 BCOLOR subcommand
 for DRAWBOX command 209
 bevel connection, definition of 395
 BLACK subcommand
 for DEFINE COLOR command 200
 BOLD
 border thickness 43
 rule thickness 38
 boldface, definition of 395
 border thickness, definition of 395
 border weight, definition of 395
 border, definition of 395
 borders 11
 DRAWBOX 43, 168
 DRAWCIRCLE 56, 174
 thickness 43
 BOTTOM, defining a circle 56
 BOX subcommand
 for DRAWGRAPHIC command 237
 boxes
 defining 41
 drawing 41
 size 42
 with dotted or dashed borders 168
 with text 100

C

CBDUMP command
 NOTRACE subcommand 194
 syntax diagram 194
 TRACE subcommand 194
 TRACEALL subcommand 194
 CENTER
 OBJECT subcommand
 for PLACE command 276
 text positioning in DRAWBOX 76
 character 106
 character box 13
 character box, definition of 395
 character set, definition of 395

character, definition of 395
 characters per inch (CPI) 139
 CHARSET subcommand
 for FONT command (MVS) 259
 for FONT command (VM) 260
 for FONT command (VSE)) 262
 CIELAB subcommand
 for DEFINE COLOR command 200
 circle radius 55
 CIRCLE subcommand
 for DRAWGRAPHIC command 236
 circles 53
 defining 55
 dotted and dashed 174
 positioning 55
 CLOSE subcommand
 for DRAWPATH command 247
 closed paths, definition of 395
 CMYK subcommand
 for DEFINE COLOR command 199
 CMYKEURO subcommand
 for DEFINE OBJECT command 202
 CMYKSWOP subcommand
 for DEFINE OBJECT command 202
 code page, definition of 395
 code point, definition of 395
 coded font, definition of 395
 coded overlay, definition of 395
 CODEPAGE subcommand
 for FONT command (MVS) 259
 for FONT command (VM) 260
 for FONT command (VSE)) 262
 color
 box background color 131
 cielab 131
 CMYK 131
 foreground color 131
 highlight 131
 IBM OCA 131
 RGB 131
 COLOR
 OBJECT subcommand
 for PLACE command 276
 PATTERN subcommand
 for PLACE command 279
 COLOR subcommand
 for DEFINE BARCODE command 198
 for DEFINE COLOR command 199
 for DRAWRULE command 256
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE)) 263
 COLUMN format 14
 command
 comments 25
 defaults 5, 24
 end marker 6
 keyword modification 338
 keywords 5, 24
 numbers 6
 sequence 5
 syntax 5, 191
 values 24
 writing of 5
 command line, definition of 395
 command stream, definition of 395
 commands
 CBDUMP
 syntax diagram 194
 comments 191
 CONTROL
 syntax diagram 195
 DEFINE BARCODE
 syntax diagram 197
 DEFINE COLOR
 syntax diagram 199
 DEFINE GROUP
 syntax diagram 201
 DEFINE OBJECT
 syntax diagram 202
 DEFINE PATTERN
 syntax diagram 204
 DRAWBOX
 syntax diagram 206
 DRAWCIRCLE
 syntax diagram 217
 DRAWGRAPHIC BOX
 syntax diagram 229
 DRAWGRAPHIC CIRCLE
 syntax diagram 231
 DRAWGRAPHIC ELLIPSE
 syntax diagram 233
 DRAWGRAPHIC FILLETS
 syntax diagram 235
 DRAWGRAPHIC MARKER
 syntax diagram 236
 DRAWGRAPHIC PATH
 syntax diagram 234
 DRAWMASK
 syntax diagram 242
 DRAWPATH
 syntax diagram 243
 DRAWRULE
 syntax diagram 255
 FONT (MVS)
 syntax diagram 258
 FONT (VM)
 syntax diagram 260
 FONT (VSE)
 syntax diagram 262
 numbers 191
 ORIENT
 syntax diagram 264
 OVERLAY
 syntax diagram 267
 PLACE BARCODE
 syntax diagram 269
 PLACE groups
 syntax diagram 275
 PLACE OBJECT
 syntax diagram 276

commands (*continued*)
 PLACE PATTERN
 syntax diagram 278
 PLACE segments
 syntax diagram 275
 POSITION
 syntax diagram 280
 SEGMENT (MVS)
 syntax diagram 283
 SEGMENT (VM)
 syntax diagram 284
 SEGMENT (VSE)
 syntax diagram 285
 SETTEXT
 syntax diagram 286
 SETUNITS
 syntax diagram 291
 comment, definition of 395
 comments 6
 multiple-word 25
 one-word 25
 Comments
 for overlay design 25
 writing of 25
 connection 60
 CONNECTION subcommand
 for DRAWPATH command 245, 247
 continuous forms paper, definition of 395
 CONTROL command
 ALL subcommand 195
 ERROR subcommand 195
 NOSOSI subcommand 196
 NOSTORE subcommand 195
 NOSUMMARY subcommand 195
 REPLACE subcommand 195
 SOSI subcommand 196
 STORE subcommand 195
 SUMMARY subcommand 195
 syntax diagram 195
 WARN subcommand 195
 control words
 reading syntax diagrams 192
 coordinate 9
 coordinate, definition of 396
 corner arc, definition of 396
 corner length, definition of 396
 CORNERLENGTH subcommand
 for SETUNITS command 293
 CPI, definition of 395
 cut-sheet paper, definition of 396

D

DARK
 shading in DRAWCIRCLE 224
 DASHED
 rule type 38
 data definition statement, definition of 396
 data set, definition of 396
 data-set allocation 305

DATAMATRIX BARCODE parameter
 for PLACE command 270
 DBCS
 in SETTEXT 289
 DDNAME *name* subcommand
 for FONT command (MVS) 259
 for SEGMENT command (MVS) 283
 DDNAME FONTDD subcommand
 for FONT command (MVS) 259
 DDNAME SEGDD subcommand
 for SEGMENT command (MVS) 283
 DDname, definition of 396
 default units of measurements 138
 default value, definition of 396
 defaults 5, 51
 SETUNITS 137
 DEFINE BARCODE command
 modifier subcommand 198
 name subcommand 197
 type subcommand 197
 COLOR subcommand 198
 FONT subcommand 198
 HEIGHT subcommand 198
 MODWIDTH subcommand 198
 RATIO subcommand 198
 syntax diagram 197
 DEFINE COLOR command
 colorname subcommand 199
 BLACK subcommand 200
 CIELAB subcommand 200
 CMYK subcommand 199
 COLOR subcommand 199
 HIGHLIGHT subcommand 200
 OCA subcommand 199
 RGB subcommand 199
 syntax diagram 199
 DEFINE GROUP command
 command word subcommand 201
 group commands subcommand 201
 section name subcommand 201
 section type subcommand 201
 syntax diagram 201
 DEFINE OBJECT command
 line coding subcommand 204
 name subcommand 202
 pattern type subcommand 204
 section name subcommand 204
 section type subcommand 204
 CMYKEURO subcommand 202
 CMYKSWOP subcommand 202
 KEEP subcommand 202
 NOKEEP subcommand 202
 OBJECT subcommand 202
 OBNAME subcommand 202
 OBTYPe subcommand 202
 syntax diagram 202
 DEFINE PATTERN command
 syntax diagram 204
 descender 99
 descender, definition of 396
 DIAGONAL 167

DIAGONAL subcommand
 for DRAWBOX command 220
 disjointed path 254
 disjointed path, definition of 396
 DOTTED
 rule type 38
 double-byte character set, definition of 396
 double-byte coded font, definition of 396
 double-byte font, definition of 396
 DRAWBOX
 balanced text
 blanks in 96
 border thickness 43
 border type 43
 box height 41
 box width 41
 DIAGONAL 167
 line spacing 97
 placement 86
 positioning boxes 41
 positioning text 76
 repeating boxes 154
 size 42
 SOSI mode 70
 width 42
 DRAWBOX command
 axis selection subcommand
 text 220
 border thickness subcommand 208, 219
 border type subcommand 208, 219
 box height subcommand 208
 box width subcommand 207
 circle portion subcommand 219
 circle radius subcommand 218
 colorname subcommand 209
 corner selection subcommand 209
 diagonal selection subcommand 209, 220
 direction subcommand
 spaced repetition 210
 horizontal coordinate subcommand
 location repetition 211
 repetitions subcommand
 spaced repetition 210
 shade area subcommand 212
 shade pattern subcommand 212
 shade type subcommand 212
 spacing value subcommand
 spaced repetition 210
 spacing word subcommand
 spaced repetition 210
 vertical coordinate subcommand
 location repetition 211
 AXIS subcommand 220
 BCOLOR subcommand 209
 DIAGONAL subcommand 220
 LOCATION subcommand
 location repetition 211
 REPEAT subcommand
 location repetition 211
 spaced repetition 210
 SHADE *colorname* subcommand 212
 DRAWBOX command (*continued*)
 SHADE subcommand 211
 spaced repetition 209
 syntax diagram 206
 WITHTEXT subcommand 213
 box 213
 font names 214
 format 213
 line spacing 214
 orientation 213
 placement 213
 text type 215
 text 215
 underlining 215
 SOSI mode 215
 DRAWCIRCLE
 adding text 106
 border thickness 56
 border type 56
 circle portion 56
 circle radius 55
 diagonal 173
 repeating circles 175
 WITHTEXT
 LINE 226
 DRAWCIRCLE command
 direction subcommand
 spaced repetition 221
 horizontal coordinate subcommand
 location repetition 223
 location option subcommand
 location repetition 223
 repetitions subcommand
 spaced repetition 221
 shade area subcommand
 location repetition 224
 shade pattern subcommand
 location repetition 224
 shade type subcommand
 location repetition 224
 spacing value subcommand
 spaced repetition 221
 vertical coordinate subcommand
 location repetition 223
 location repetition 223
 REPEAT subcommand
 location repetition 223
 spaced repetition 221
 SHADE subcommand
 location repetition 224
 spaced repetition 221
 SPACED subcommand
 spaced repetition 221
 syntax diagram 217
 WITHTEXT subcommand 225
 circle 225
 font names 226
 format 225
 line spacing 226
 orientation 225
 placement 225

DRAWCIRCLE command (*continued*)
 WITHTEXT subcommand (*continued*)
 text type 227
 text 227
 underlining 227
 SOSI mode 227

DRAWGRAPHIC BOX command
 syntax diagram 229

DRAWGRAPHIC CIRCLE command
 syntax diagram 231

DRAWGRAPHIC command
colorname subcommand 238
fillpattern subcommand 238
linetype subcommand 238
linewidth subcommand 238
marker symbol subcommand 238
 BOX subcommand 237
 CIRCLE subcommand 236
 ELLIPSE subcommand 236
 FILLETS subcommand 237
 MARKER subcommand 238
 PATH subcommand 237
 programming samples 241
 REPEAT subcommand 240
 WITHTEXT subcommand 240

DRAWGRAPHIC ELLIPSE command
 syntax diagram 233

DRAWGRAPHIC FILLETS command
 syntax diagram 235

DRAWGRAPHIC MARKER command
 syntax diagram 236

DRAWGRAPHIC PATH command
 syntax diagram 234

drawing
 boxes 41
 circles 53
 paths 58, 179, 243
 rules 36, 149, 255

DRAWMASK
 and SETUNITS 147
 command word 33
 horizontal spacing 33
 vertical spacing 33

DRAWMASK command
first spacing subcommand 242
second spacing subcommand 242

HCOLOR subcommand 242
 syntax diagram 242

VCOLOR subcommand 242

DRAWPATH
 close option 62
 connection type 60
 origin option 61, 247
 path end shape 181
 path start shape 181
 path thickness 60
 path type 60
 shading paths 187

DRAWPATH command
connection length subcommand 245, 248
connection type subcommand 245, 247

DRAWPATH command (*continued*)
direction subcommand 249
first coordinate subcommand 247
horizontal coordinate subcommand
 location repetition 250
origin option subcommand 247
path end shape subcommand 245
path start shape subcommand 244
path thickness subcommand 244
path type subcommand 244
repetitions subcommand 249
second coordinate subcommand 247
spacing value subcommand 249
vertical coordinate subcommand
 location repetition 250
 CLOSE subcommand 247
 CONNECTION subcommand 245, 247
 location repetition 250
 LOCATION subcommand
 location repetition 250
 PATHEND subcommand 244
 REPEAT subcommand 251
 location repetition 250
 spaced repetition 249
 SHADE subcommand
 path 251
 shade pattern 251
 shade type 251
 spaced repetition 249
 SPACED subcommand
 spaced repetition 249
 specifying path points 247
 syntax diagram 243
 tips
 shading closed paths 253
 using MITER connections 252
 using ROUNDED connections 252
 TO subcommand 247

DRAWRULE
 repeating rules 149
 type 38

DRAWRULE command
direction subcommand
 spaced repetition 256
horizontal coordinate subcommand
 location repetition 257
location option subcommand
 location repetition 257
repetitions subcommand
 spaced repetition 256
rule direction subcommand 255
rule length subcommand
 path 255
rule thickness subcommand
 path 255
rule type subcommand
 path 255
spacing value subcommand
 spaced repetition 256
vertical coordinate subcommand
 location repetition 257

DRAWRULE command (*continued*)

- COLOR subcommand
 - path* 256
 - location repetition 257
 - REPEAT subcommand
 - location repetition 257
 - spaced repetition 256
 - spaced repetition 256
 - SPACED subcommand
 - spaced repetition 256
 - syntax diagram 255
- duplex, definition of 396

E

- electronic overlay, definition of 396
- ELLIPSE subcommand
 - for DRAWGRAPHIC command 236
- ENCODED 121
 - coding 121
- encoded, definition of 396
- end marker, definition of 396
- entry, definition of 396
- error
 - messages 345
- ERROR 26
- error message, definition of 396
- ERROR subcommand
 - for CONTROL command 195
- error tracing
 - use of CBDUMP 194

F

- features of OGL/370 137, 171
- FILETYPE *name* subcommand
 - for FONT command (VM) 261
 - for SEGMENT command (VM) 284
- FILETYPE FONT38PP subcommand
 - for FONT command (VM) 261
- FILETYPE PSEG38PP subcommand
 - for SEGMENT command (VM) 284
- FILL
 - in DRAWGRAPHIC command 229
 - OBJECT subcommand
 - for PLACE command 276
- FILLETS subcommand
 - for DRAWGRAPHIC command 237
- fixed-space font, definition of 396
- font character set, definition of 396
- FONT command (MVS)
 - font name* subcommand 258
 - member ID* subcommand 258
 - CHARSET subcommand 259
 - CODEPAGE 259
 - COLOR subcommand 259
 - DDNAME *name* subcommand 259
 - DDNAME FONTDD subcommand 259
 - HEIGHT subcommand
 - for FONT command (MVS) 259
 - SCALE subcommand 259

FONT command (MVS) (*continued*)

- syntax diagram 258
- UCOLOR subcommand 259
- FONT command (VM)
 - font name* subcommand 260
 - member ID* subcommand 260
 - CHARSET subcommand 260
 - CODEPAGE subcommand 260
 - COLOR subcommand 261
 - FILETYPE *name* subcommand 261
 - FILETYPE FONT38PP subcommand 261
 - HEIGHT subcommand 261
 - SCALE subcommand 261
 - syntax diagram 260
 - UCOLOR subcommand 261
- FONT command (VSE)
 - font name* subcommand 262
 - member ID* subcommand 262
 - orientation* subcommand 264
 - CHARSET subcommand 262
 - CODEPAGE subcommand 262
 - COLOR subcommand 263
 - HEIGHT subcommand 263
 - SCALE subcommand 263
 - syntax diagram 262
 - UCOLOR subcommand 263
- FONT subcommand
 - for DEFINE BARCODE command 198
- font, definition of 396
- fonts 13
- form map, definition of 397
- form, definition of 397
- format
 - COLUMN 14
 - MODERN 14
 - TATE 15
- format, definition of 397
- FORMDEF 16
- forms flash, definition of 397
- forms overlay, definition of 397
- frame, definition of 397

G

- GDDM, definition of 397
- GOCA 229
- graphic, definition of 397
- Graphical Data Display Manager 16
- Graphical Object Content Architecture 229
- graphics 15, 117
 - commands 128
 - image patterns 117
 - page segments 117
- grid 33
- group 15
 - using SETUNITS inside 164
- GROUP subcommand
 - for PLACE command 275
- group, definition of 397

H

HALF
 circle shape 56
 corner length value 143
hardware requirements 16
HCOLOR subcommand
 for DRAWMASK command 242
HEIGHT subcommand
 for DEFINE BARCODE command 198
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE)) 263
HIGHLIGHT subcommand
 for DEFINE COLOR command 200
Hiragana, definition of 397
HRI position, BARCODE subcommand
 for PLACE command 269

I

IBM 5550 Work Station 320
IBM AFP Printer 17
ID name, definition of 397
ID, definition of 397
image, definition of 397
informational message, definition of 397
invisible boxes 101
invocation, definition of 397
IPO 8

J

JCL, definition of 397
JCS, definition of 397
Job Control Language (JCL)
 submitting overlays for processing
 MVS 307
 VM 312
 VSE 309
JUSTIFY 87
justify, definition of 397

K

Kanji
 description of 14
 use of 320
Kanji, definition of 397
KEEP subcommand
 for DEFINE OBJECT command 202
kerning 99
kerning, definition of 397
keyword, definition of 397
keywords
 abbreviations 338
 list of 337
 modification 338
 requirements 5, 24

L

LARGE, corner length value 143
LASTNO 87, 91
LEFT
 OBJECT subcommand
 for PLACE command 276
LEFT, defining a circle 56
librarian in VSE, definition of 397
library, definition of 397
LIGHT
 border thickness 43
 rule thickness 38
line printer, definition of 398
line space, definition of 398
line spacing, boxes 97
line spacing, circles 98
LINE subcommand
 for SETTEXT command 289
line, definition of 398
lines per inch (LPI) 97
LINESP 141
LOCATION 158, 176, 186
location repetition
 for DRAWCIRCLE command 223
 for DRAWPATH command 250
 for DRAWRULE command 257
location repetition, definition of 398
LOCATION subcommand
 for DRAWBOX command
 location repetition 211
 for DRAWPATH command
 location repetition 250
logical page, definition of 6, 398
LPI, definition of 398

M

manuals, related 19
MARKER subcommand
 for DRAWGRAPHIC command 238
mask, definition of 398
masks 33
MAX, corner length value 143
MAXICODE BARCODE parameter
 for PLACE command 271
maximum
 ascender 99
 descender 99
maximum ascender, definition of 398
maximum descender, definition of 398
measurement units 191, 343
MEDIUM
 border thickness 43
 corner length value 143
 rule thickness 38
medium overlay 6, 30
member ID, definition of 398
messages
 error 345
 options 26

messages (*continued*)
 types 6
 MIRROR 127
 PATTERN subcommand
 for PLACE command 279
 mirror image, definition of 398
 miter connection, definition of 398
 MITER connections
 tips
 with DRAWPATH command 252
 MODERN format 14
 MODWIDTH subcommand
 for DEFINE BARCODE command 198
 multiple-word comments 25, 191

N

NEGATIVE 127
 PATTERN subcommand
 for PLACE command 279
 negative image, definition of 398
 NOKEEP subcommand
 for DEFINE OBJECT command 202
 NOMIRROR 127
 PATTERN subcommand
 for PLACE command 279
 NONEGATIVE 127
 PATTERN subcommand
 for PLACE command 279
 normal duplex, definition of 398
 NOSOSI subcommand
 for CONTROL command 196
 NOSTORE 26
 NOSTORE subcommand
 for CONTROL command 195
 NOSUMMARY 27
 NOSUMMARY subcommand
 for CONTROL command 195
 notices 391
 Notices section 391
 NOTRACE subcommand
 for CBDUMP command 194
 NOUNDERLINE 106
 NOUNDERLINE subcommand
 for SETTEXT command 290
 numbers
 specifying 6
 numbers, specifying 191

O

OBJECT subcommand
 depth
 for PLACE command 276
 orientation
 for PLACE command 276
 width
 for PLACE command 276
 x offset
 for PLACE command 276

OBJECT subcommand (*continued*)
 y offset
 for PLACE command 276
 CENTER
 for PLACE command 276
 COLOR
 for PLACE command 276
 FILL
 for PLACE command 276
 for DEFINE OBJECT command 202
 LEFT
 for PLACE command 276
 REPEAT
 for PLACE command 276
 SCALE
 for PLACE command 276
 TRIM
 for PLACE command 276
 OBNAMe subcommand
 for DEFINE OBJECT command 202
 OBTYPe subcommand
 for DEFINE OBJECT command 202
 OCA subcommand
 for DEFINE COLOR command 199
 offset 31
 OFFSET 5
 OFFSET subcommand
 for OVERLAY command 267
 offset, definition of 398
 one-word comments 25, 191
 open paths, definition of 398
 option, definition of 398
 optional parameters
 for syntax diagrams 193
 Orient
 command word 31
 orientation 31
 ORIENT
 description of 28, 31
 ORIENT command
 syntax diagram 264
 orientation
 concept 12
 orientation, definition of 398
 origin 9
 origin, definition of 398
 output listings for overlay RCPT
 MVS 303
 VM 295
 VSE 304
 overlay 4
 definition 5
 OVERLAY
 coordinates 28
 description of 4, 27
 height 28
 name 27
 subcommand word 28
 vertical coordinates 28
 width 28

- OVERLAY command
 - horizontal coordinate* subcommand 268
 - overlay height* subcommand 267
 - overlay name* subcommand 267
 - overlay width* subcommand 267
 - vertical coordinate* subcommand 268
- OFFSET subcommand 267
- SIZE subcommand 267
- syntax diagram 267
- overlay design
 - comments 25
 - defaults 24
 - keywords 24
 - values 24
- Overlay Generation Language
 - borders 11, 168, 174
 - color 131
 - box background color 131
 - cielab 131
 - CMYK 131
 - foreground color 131
 - highlight 131
 - IBM OCA 131
 - RGB 131
 - commands
 - defaults 5
 - end marker 6
 - keywords 5, 191
 - numbers 6, 191
 - sequence 5
 - syntax 5, 191
 - writing of 5
 - comments 6, 191
 - concepts of 4
 - coordinates 9
 - definition 4
 - fonts 13
 - graphics 15, 117
 - groups 15
 - keywords 337
 - messages 345
 - all 6
 - error 6
 - informational 6
 - warning 6
 - OFFSET 5
 - orientation 12
 - origin 9
 - OVERLAY 4
 - overlay definition 5
 - page segment 15, 117
 - patterns 15, 120
 - pels 10
 - RCPT 4
 - rules 11
 - SIZE 5
 - text format 14
- overlay, definition of 398

P

- page
 - logical 6
 - physical 6
- page overlay 6
- page printer, definition of 399
- page segment 15
- page segment, definition of 399
- PAGEDEF 16
- paper origin, definition of 399
- parse, definition of 399
- path points, specifying
 - with DRAWPATH command 247
- PATH subcommand
 - for DRAWGRAPHIC command 237
- path, definition of 399
- PATHEND subcommand
 - for DRAWPATH command 244
- pattern
 - as differs from segment 117
 - definition 15
 - positioning 46
- PATTERN subcommand
 - orientation*
 - for PLACE command 278
 - section name*
 - for PLACE command 278
 - shade pattern*
 - for PLACE command 278
 - shade type*
 - for PLACE command 278
 - for PLACE command 278
 - SHADE
 - for PLACE command 278
- patterns
 - coding pels (DEFINE command) 119, 120
 - mirror patterns 127
 - modifications (PLACE command) 126
 - negative option 127
 - orientation 126
 - placing 125, 278
 - positioning 125, 129
 - sample DEFINE PATTERN command 124
 - sample pel pattern 121
 - shade pattern 126
 - shading 126
- PDF417 BARCODE parameter
 - for PLACE command 272
- pel, definition of 399
- pels 10
 - coding 121
- physical page, definition of 6
- pitch, definition of 399
- PLACE
 - command word 125
 - description of 125
 - PATTERN 125
 - pattern modifications
 - mirror option 127
 - negative option 127
 - orientation 126

PLACE (*continued*)
 pattern modifications (*continued*)
 shade type 126
 shading option 126
 section type 125
 SEGID 125

PLACE command
section name subcommand 275
section type 278
section type subcommand 275
 additional entries for placing patterns 278
 BARCODE *name* subcommand 269
 BARCODE *orientation* subcommand 269
 BARCODE *text type* subcommand 269
 BARCODE 2DPARMS subcommand 269
 BARCODE ASTERISK subcommand 269
 BARCODE HRI position subcommand 269
 BARCODE subcommand 269
 syntax diagram 269
 DATAMATRIX BARCODE parameter 270
 GROUP subcommand 275
 groups
 syntax diagram 275
 MAXICODE BARCODE parameter 271
 OBJECT
 syntax diagram 276
 OBJECT subcommand
name 276
 PATTERN
 syntax diagram 278
 PATTERN subcommand 278
mirror option 279
negative option 279
 COLOR 279
 MIRROR 279
 NEGATIVE 279
 NOMIRROR 279
 NONEGATIVE 279
 PDF417 BARCODE parameter 272
 SEGID subcommand 275
 segments
 syntax diagram 275

placement
 BALANCE 86, 91
 JUSTIFY 87, 91
 of text 76, 86

placing text 86

placing text in overlay
 syntax rules 191

point size, definition of 399

point, definition of 399

POSITION
 boxes 41
 circles 55
 description of 37
 first coordinate 37
 graphics 125
 origin option 37
 paths 59
 second coordinate 38
 with SETUNITS 144

POSITION command
absolute positioning subcommand 280
first coordinate subcommand 280
origin option subcommand 280
relative positioning subcommand 280
second coordinate subcommand 280
 syntax diagram 280

POSITIONING subcommand
 for SETUNITS command 293

PPFA
 description of 16

PPFA, definition of 399

preprinted form, definition of 399

primary default, definition of 399

print line, definition of 399

Print Management Facility
 considerations 315
 description of 16

Print Management Facility (PMF), definition of 399

print position, definition of 399

Print Services Access Facility, definition of 399

Print Services Facility 4

Print Services Facility, definition of 399

printable area 31

printable area, definition of 399

printers 18

printing
 overlays 335
 printers 335
 subsystems
 characteristics 335

program invocation
 in MVS 307
 in VM 312
 in VSE 309

programming samples
 for DRAWGRAPHIC command 241

publications, related 19

Q

QUARTER, circle shape 57

R

raster pattern overlay, definition of 399

raster pattern, definition of 399

RATIO subcommand
 for DEFINE BARCODE command 198

RCPT 4

reading syntax diagrams 192

record, definition of 399

related manuals 19

related publications 19

relative orientation 13

RELATIVE positioning
 first coordinate 46
 second coordinate 46

relative positioning, definition of 400

REPEAT
 for boxes
 description of 154, 155
 horizontal coordinate 158
 location option 158
 repetitions 156
 spacing value 156
 spacing word 156
 vertical coordinate 158
 for circles
 description of 175
 horizontal coordinate 176
 location option 176
 repetitions 176
 spacing value 176
 spacing word 176
 vertical coordinate 177
 for paths
 description of 179, 186
 location 186
 spacing value 186
 for rules
 description of 149
 horizontal coordinate 152
 location option 152
 repetitions 150
 spacing value 150
 spacing word 150
 vertical coordinate 152
 OBJECT subcommand
 for PLACE command 276
REPEAT subcommand
 for DRAWBOX command
 location repetition 211
 spaced repetition 210
 for DRAWCIRCLE command
 location repetition 223
 spaced repetition 221
 for DRAWGRAPHIC command 240
 for DRAWPATH command
 location repetition 250
 spaced repetition 249
 for DRAWRULE command
 location repetition 257
 spaced repetition 256
 repeating parameters
 for syntax diagrams 193
REPLACE 26
REPLACE subcommand
 for CONTROL command 195
 required parameters
 for syntax diagrams 192
RGB subcommand
 for DEFINE COLOR command 199
RIGHT, defining a circle 56
 rotated font, definition of 400
 rotation, definition of 400
 rounded connection, definition of 400
ROUNDED connections 183
 tips
 with DRAWPATH command 252

rule spacing 33, 256
 rule, definition of 400
 rules 11
 defining 38
 drawing 36
 positioning 37

S

samples, programming
 for DRAWGRAPHIC command 241
SBCS
 in WITHTEXT 289
SCALE
 OBJECT subcommand
 for PLACE command 276
SCALE subcommand
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE)) 263
 screen, definition of 400
 secondary default, definition of 400
SEGID subcommand
 for PLACE command 275
SEGMENT
 DD statement name 118
 DD statement word 118
 member ID 118
 segment name 117
SEGMENT command (MVS)
 member ID subcommand 283
 name subcommand 283
 DDNAME *name* subcommand 283
 DDNAME SEGDD subcommand 283
 syntax diagram 283
SEGMENT command (VM)
 member ID subcommand 284
 name subcommand 284
 FILETYPE *name* subcommand 284
 FILETYPE PSEG38PP subcommand 284
 syntax diagram 284
SEGMENT command (VSE)
 member ID subcommand 285
 name subcommand 285
 syntax diagram 285
SETTEXT
 alignment 109
 description of 108
 font names 111
 format 109
 LINE 111
 line spacing 111
 orientation 109
 restrictions 109
 SOSI mode 111
 text 112
 text format 109
 underlining 112
SETTEXT command
 alignment subcommand 287
 font names subcommand 289

SETTEXT command (*continued*)
format subcommand 286
line spacing subcommand 288
orientation subcommand 286
text type subcommand 290
text subcommand 290
LINE subcommand 289
NOUNDERLINE subcommand 290
SOSI1 subcommand 289
SOSI2 subcommand 290
 syntax diagram 286
UNDERLINE subcommand 290

SETUNITS
 and **DRAWMASK** 147
 box positioning 144
CORNERLENGTH option 140, 141
 function 137
 linespace option 140
 linespace value 140
 primary default 139
 rule positioning 144
 secondary default 140

SETUNITS command
length value subcommand 293
linespace option subcommand 292
linespace value subcommand 292
positioning option subcommand 293
primary default subcommand 291
secondary default subcommand 292
CORNERLENGTH subcommand 293
POSITIONING subcommand 293
 syntax diagram 291
TEXTMARGIN subcommand 293

shade
 boxes with dotted or dashed borders 168
 patterns 18
 percentages
 standard shade pattern 326
 using **DRAWBOX** 159
 varying
 description of 160

SHADE
PATTERN subcommand
 for **PLACE** command 278

SHADE *colorname* subcommand
 for **DRAWBOX** command 212

SHADE subcommand
 for **DRAWBOX** command 211
 for **DRAWCIRCLE** command
 location repetition 224
 for **DRAWPATH** command 251
 path 251
 shade pattern 251
 shade type 251

shade, definition of 400

shading
 in **DRAWBOX** 44
 in **DRAWCIRCLE** 57
 screen pattern 44

 shading closed paths
 tips
 with **DRAWPATH** command 253

 shift-in (SI) character 27

 shift-in character, definition of 400

 shift-out (SO) character 27

 shift-out character, definition of 400

 simplex, definition of 400

 single-byte character set, definition of 400

 single-byte coded font, definition of 400

 single-byte font
 proportional 14
 uniformly spaced 14

SIZE 5

SIZE subcommand
 for **OVERLAY** command 267

SMALL, corner length value 142

software requirements 16

SOLID
 border type 56
 rule type 38

sosi option 27

SOSI subcommand
 for **CONTROL** command 196

SOSI1 subcommand
 for **SETTEXT** command 289

SOSI2 subcommand
 for **SETTEXT** command 290

source listing, definition of 400

spaced repetition
 for boxes 156
 for circles 186
 for **DRAWBOX** command 209
 for **DRAWCIRCLE** command 221
 for **DRAWPATH** command 249
 for **DRAWRULE** command 256
 for rules 150, 152
 for varying shading 187
 for varying shading and text 159, 177

spaced repetition, definition of 400

SPACED subcommand
 for **DRAWCIRCLE** command
 spaced repetition 221
 for **DRAWPATH** command
 spaced repetition 249
 for **DRAWRULE** command
 spaced repetition 256

special functions
 BARCODE DATAMATRIX subcommand 271

specifying path points
 with **DRAWPATH** command 247

storage
 limitations 335

STORE 26

STORE subcommand
 for **CONTROL** command 195

string, definition of 400

style rules
 for syntax diagrams 192

subcommand, definition of 400

subcommands

- absolute positioning* 280
- alignment* 287
- axis selection*
 - for DRAWBOX command 220
- border thickness*
 - for DRAWBOX command 208, 219
- border type*
 - for DRAWBOX command 208, 219
- box height*
 - for DRAWBOX command 208
- box width*
 - for DRAWBOX command 207
- circle portion*
 - for DRAWBOX command 219
- circle radius*
 - for DRAWBOX command 218
- colorname*
 - for DEFINE COLOR command 199
 - for DRAWBOX command 209
 - for DRAWGRAPHIC command 238
- command word*
 - for DEFINE GROUP command 201
- connection length*
 - for DRAWPATH command 245, 248
- connection type*
 - for DRAWPATH command 245, 247
- corner selection*
 - for DRAWBOX command 209
- diagonal selection*
 - for DRAWBOX command 209, 220
- direction*
 - for DRAWPATH command 249
- direction (spaced repetition)*
 - for DRAWBOX command 210
 - for DRAWCIRCLE command 221
 - for DRAWRULE command 256
- fillpattern*
 - for DRAWGRAPHIC command 238
- first coordinate* 280
 - for DRAWPATH command 247
- first spacing*
 - for DRAWMASK command 242
- font name*
 - for FONT command (MVS) 258
 - for FONT command (VM) 260
 - for FONT command (VSE) 262
- font names* 289
- format* 286
- group commands*
 - for DEFINE GROUP command 201
- horizontal coordinate*
 - for OVERLAY command 268
- horizontal coordinate (location repetition)*
 - for DRAWBOX command 211
 - for DRAWCIRCLE command 223
 - for DRAWPATH command 250
 - for DRAWRULE command 257
- length value*
 - for SETUNITS command 293

subcommands (*continued*)

- line coding*
 - for DEFINE OBJECT command 204
- line spacing* 288
- linespace option*
 - for SETUNITS command 292
- linespace value*
 - for SETUNITS command 292
- linetype*
 - for DRAWGRAPHIC command 238
- linewidth*
 - for DRAWGRAPHIC command 238
- location option (location repetition)*
 - for DRAWCIRCLE command 223
 - for DRAWRULE command 257
- marker symbol*
 - for DRAWGRAPHIC command 238
- member ID* 283, 284, 285
 - for FONT command (MVS) 258
 - for FONT command (VM) 260
 - for FONT command (VSE) 262
- modifier*
 - for DEFINE BARCODE command 198
- name* 283, 284, 285
 - for DEFINE BARCODE command 197
 - for DEFINE OBJECT command 202
- orientation* 286
 - for FONT command (VSE) 264
- origin option* 280
 - for DRAWPATH command 247
- overlay height*
 - for OVERLAY command 267
- overlay name*
 - for OVERLAY command 267
- overlay width*
 - for OVERLAY command 267
- path end shape*
 - for DRAWPATH command 245
- path start shape*
 - for DRAWPATH command 244
- path thickness*
 - for DRAWPATH command 244
- path type*
 - for DRAWPATH command 244
- pattern type*
 - for DEFINE OBJECT command 204
- positioning option*
 - for SETUNITS command 293
- primary default*
 - for SETUNITS command 291
- relative positioning* 280
- repetitions*
 - for DRAWPATH command 249
- repetitions (spaced repetition)*
 - for DRAWBOX command 210
 - for DRAWCIRCLE command 221
 - for DRAWRULE command 256
- rule direction*
 - for DRAWRULE command 255
- rule length*
 - for DRAWRULE command 255

subcommands (*continued*)

rule thickness
 for DRAWRULE command 255

rule type
 for DRAWRULE command 255

second coordinate 280
 for DRAWPATH command 247

second spacing
 for DRAWMASK command 242

secondary default
 for SETUNITS command 292

section name 275
 for DEFINE GROUP command 201
 for DEFINE OBJECT command 204

section type 275
 for DEFINE GROUP command 201
 for DEFINE OBJECT command 204

shade area
 for DRAWBOX command 212

shade area (location repetition)
 for DRAWCIRCLE command 224

shade pattern
 for DRAWBOX command 212

shade pattern (location repetition)
 for DRAWCIRCLE command 224

shade type
 for DRAWBOX command 212

shade type (location repetition)
 for DRAWCIRCLE command 224

spacing value
 for DRAWPATH command 249

spacing value (spaced repetition)
 for DRAWBOX command 210
 for DRAWCIRCLE command 221
 for DRAWRULE command 256

spacing word (spaced repetition)
 for DRAWBOX command 210

text type 290

text 290

type
 for DEFINE BARCODE command 197

vertical coordinate
 for OVERLAY command 268

vertical coordinate (location repetition)
 for DRAWBOX command 211
 for DRAWCIRCLE command 223
 for DRAWPATH command 250
 for DRAWRULE command 257

ALL
 for CONTROL command 195

AXIS
 for DRAWBOX command 220

BARCODE
 for PLACE command 269

BARCODE *name*
 for PLACE command 269

BARCODE *orientation*
 for PLACE command 269

BARCODE *text type*
 for PLACE command 269

subcommands (*continued*)

BARCODE 2DPARMS
 for PLACE command 269

BARCODE ASTERISK
 for PLACE command 269

BARCODE HRI position
 for PLACE command 269

BCOLOR
 for DRAWBOX command 209

BLACK
 for DEFINE COLOR command 200

BOX
 for DRAWGRAPHIC command 237

CHARSET
 for FONT command (MVS) 259
 for FONT command (VM) 260
 for FONT command (VSE) 262

CIELAB
 for DEFINE COLOR command 200

CIRCLE
 for DRAWGRAPHIC command 236

CLOSE
 for DRAWPATH command 247

CMYK
 for DEFINE COLOR command 199

CMYKEURO
 for DEFINE OBJECT command 202

CMYKSWOP
 for DEFINE OBJECT command 202

CODEPAGE
 for FONT command (MVS) 259
 for FONT command (VM) 260
 for FONT command (VSE) 262

COLOR
 for DEFINE BARCODE command 198
 for DEFINE COLOR command 199
 for DRAWRULE command 256
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE) 263

CONNECTION
 for DRAWPATH command 245, 247

CORNERLENGTH
 for SETUNITS command 293

DDNAME *name* 283
 for FONT command (MVS) 259

DDNAME FONTDD
 for FONT command (MVS) 259

DDNAME SEGDD 283

DIAGONAL
 for DRAWBOX command 220

ELLIPSE
 for DRAWGRAPHIC command 236

ERROR
 for CONTROL command 195

FILETYPE *name* 284
 for FONT command (VM) 261

FILETYPE FONT38PP
 for FONT command (VM) 261

FILETYPE PSEG38PP 284

subcommands (*continued*)

FILLETS
 for DRAWGRAPHIC command 237
 FONT
 for DEFINE BARCODE command 198
 for PLACE command 275, 276, 278, 279
 for POSITION command 280
 for SEGMENT command (MVS) 283
 for SEGMENT command (VM) 284
 for SEGMENT command (VSE) 285
 for SETTEXT command 286, 287, 288, 289, 290
 GROUP 275
 HCOLOR
 for DRAWMASK command 242
 HEIGHT
 for DEFINE BARCODE command 198
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE) 263
 HIGHLIGHT
 for DEFINE COLOR command 200
 KEEP
 for DEFINE OBJECT command 202
 LINE 289
 LOCATION (location repetition)
 for DRAWBOX command 211
 for DRAWPATH command 250
 MARKER
 for DRAWGRAPHIC command 238
 MODWIDTH
 for DEFINE BARCODE command 198
 NOKEEP
 for DEFINE OBJECT command 202
 NOSOSI
 for CONTROL command 196
 NOSTORE
 for CONTROL command 195
 NOSUMMARY
 for CONTROL command 195
 NOTRACE
 for CBDUMP command 194
 NOUNDERLINE 290
 OBJECT
 for DEFINE OBJECT command 202
 OBJECT *depth* 276
 OBJECT *name* subcommand 276
 OBJECT *orientation* 276
 OBJECT *width* 276
 OBJECT *x offset* 276
 OBJECT *y offset* 276
 OBJECT CENTER 276
 OBJECT COLOR 276
 OBJECT FILL 276
 OBJECT LEFT 276
 OBJECT REPEAT 276
 OBJECT SCALE 276
 OBJECT TRIM 276
 OBNAM
 for DEFINE OBJECT command 202
 OBTYPE
 for DEFINE OBJECT command 202

subcommands (*continued*)

OCA
 for DEFINE COLOR command 199
 OFFSET
 for OVERLAY command 267
 PATH
 for DRAWGRAPHIC command 237
 PATHEND
 for DRAWPATH command 244
 PATTERN 278
 PATTERN *mirror option* 279
 PATTERN *negative option* 279
 PATTERN *orientation* 278
 PATTERN *section name* 278
 PATTERN *shade pattern* 278
 PATTERN *shade type* 278
 PATTERN COLOR 279
 PATTERN MIRROR 279
 PATTERN NEGATIVE 279
 PATTERN NOMIRROR 279
 PATTERN NONEGATIVE 279
 PATTERN SHADE 278
 POSITIONING
 for SETUNITS command 293
 RATIO
 for DEFINE BARCODE command 198
 REPEAT
 for DRAWGRAPHIC command 240
 REPEAT (location repetition)
 for DRAWBOX command 211
 for DRAWCIRCLE command 223
 for DRAWPATH command 250
 for DRAWRULE command 257
 REPEAT (spaced repetition)
 for DRAWBOX command 210
 for DRAWCIRCLE command 221
 for DRAWPATH command 249
 for DRAWRULE command 256
 REPLACE
 for CONTROL command 195
 RGB
 for DEFINE COLOR command 199
 SCALE
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE) 263
 SEGID 275
 SHADE
 for DRAWBOX command 211
 for DRAWPATH command 251
 SHADE (location repetition)
 for DRAWCIRCLE command 224
 SHADE *colorname*
 for DRAWBOX command 212
 SHADE *path*
 for DRAWPATH command 251
 SHADE *shade pattern*
 for DRAWPATH command 251
 SHADE *shade type*
 for DRAWPATH command 251

subcommands (*continued*)

SIZE
 for OVERLAY command 267

SOSI
 for CONTROL command 196

SOSI1 289

SOSI2 290

SPACED (spaced repetition)
 for DRAWCIRCLE command 221
 for DRAWPATH command 249
 for DRAWRULE command 256

STORE
 for CONTROL command 195

SUMMARY
 for CONTROL command 195

TEXTMARGIN
 for SETUNITS command 293

TO
 for DRAWPATH command 247

TRACE
 for CBDUMP command 194

TRACEALL
 for CBDUMP command 194

UCOLOR
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE) 263

UNDERLINE 290

VCOLOR
 for DRAWMASK command 242

WARN
 for CONTROL command 195

WITHTEXT
 for DRAWBOX command 213
 for DRAWCIRCLE command 225
 for DRAWGRAPHIC command 240

WITHTEXT *box*
 for DRAWBOX command 213

WITHTEXT *circle*
 for DRAWCIRCLE command 225

WITHTEXT *font names*
 for DRAWBOX command 214
 for DRAWCIRCLE command 226

WITHTEXT *format*
 for DRAWBOX command 213
 for DRAWCIRCLE command 225

WITHTEXT *line spacing*
 for DRAWBOX command 214
 for DRAWCIRCLE command 226

WITHTEXT *orientation*
 for DRAWBOX command 213
 for DRAWCIRCLE command 225

WITHTEXT *placement*
 for DRAWBOX command 213
 for DRAWCIRCLE command 225

WITHTEXT *text type*
 for DRAWBOX command 215
 for DRAWCIRCLE command 227

WITHTEXT *text*
 for DRAWBOX command 215
 for DRAWCIRCLE command 227

subcommands (*continued*)

WITHTEXT *underlining*
 for DRAWBOX command 215
 for DRAWCIRCLE command 227

WITHTEXT SOSI mode
 for DRAWBOX command 215
 for DRAWCIRCLE command 227

SUMMARY 27

SUMMARY subcommand
 for CONTROL command 195

symbolic data set, definition of 400

symbolic data sets
 description of 114, 319
 name 114
 text 114
 text type 114
 use of 319

symbols
 for syntax diagrams 192

syntax 5, 191

syntax diagram
 for CBDUMP command 194
 for CONTROL command 195
 for DEFINE BARCODE command 197
 for DEFINE COLOR command 199
 for DEFINE GROUP command 201
 for DEFINE OBJECT command 202
 for DEFINE PATTERN command 204
 for DRAWBOX command 206
 for DRAWCIRCLE command 217
 for DRAWGRAPHIC BOX command 229
 for DRAWGRAPHIC CIRCLE command 231
 for DRAWGRAPHIC ELLIPSE command 233
 for DRAWGRAPHIC FILLETS command 235
 for DRAWGRAPHIC MARKER command 236
 for DRAWGRAPHIC PATH command 234
 for DRAWMASK command 242
 for DRAWPATH command 243
 for DRAWRULE command 255
 for FONT command (MVS) 258
 for FONT command (VM) 260
 for FONT command (VSE) 262
 for ORIENT command 264
 for OVERLAY command 267
 for PLACE command
 BARCODE subcommand 269
 groups 275
 OBJECT 276
 PATTERN 278
 segments 275
 for POSITION command 280
 for SEGMENT command (MVS) 283
 for SEGMENT command (VM) 284
 for SEGMENT command (VSE) 285
 for SETTEXT command 286
 for SETUNITS command 291

syntax diagrams, optional parameters 193

syntax diagrams, reading 192

syntax diagrams, repeating parameters 193

syntax diagrams, required parameters 192

syntax diagrams, style rules 192

syntax diagrams, symbols 192
syntax rules
 abbreviations 191
 general rules 191
 making comments 191
 placing text in overlay 191
 using numbers 191
syntax, definition of 400
system programmer, definition of 400

T

TATE format 15
tate, definition of 400
text block orientation 86
text block, definition of 400
text direction restrictions 335
text format and orientation 14, 76
text margin
 in boxes 70
 in circles 73
text margin, definition of 400
text margins
 with SETUNITS 144
text origin 108
text positioning in boxes 77
text segment, definition of 400
text string, definition of 400
text, varying
 box 161
 description of 178
TEXTMARGIN 140
TEXTMARGIN subcommand
 for SETUNITS command 293
TO subcommand
 for DRAWPATH command 247
top
 defining a circle 56
 of overlay 28
 of paper 28
TOPLEFT, defining a circle 57
TOPRIGHT, defining a circle 57
TRACE subcommand
 for CBDUMP command 194
TRACEALL subcommand
 for CBDUMP command 194
tracing, use of CBDUMP for error 194
TRIM
 OBJECT subcommand
 for PLACE command 276
tumble duplex, definition of 401
two-dimensional BARCODE
 placing DATAMATRIX 270
 placing MAXICODE 271
 placing PDF417 272
typeface, definition of 401
typographic font, definition of 401

U

UCOLOR subcommand
 for FONT command (MVS) 259
 for FONT command (VM) 261
 for FONT command (VSE)) 263
UNDERLINE 106
UNDERLINE subcommand
 for SETTEXT command 290
uniformly spaced font, definition of 401
units of measure 10
units of measurements
 defaults 138
unprintable area, definition of 401

V

value, definition of 401
VCOLOR subcommand
 for DRAWMASK command 242
VM, definition of 401
VSE, definition of 401
VSE/AF, definition of 401

W

WARN 26
WARN subcommand
 for CONTROL command 195
warning message, definition of 401
WHOLE, circle shape 56
WITHTEXT subcommand
 for DRAWBOX command 213
 box 213
 font names 214
 format 213
 line spacing 214
 orientation 213
 placement 213
 text type 215
 text 215
 underlining 215
 SOSI mode 215
 for DRAWCIRCLE command 225
 circle 225
 font names 226
 format 225
 line spacing 226
 orientation 225
 placement 225
 text type 227
 text 227
 underlining 227
 SOSI mode 227
 for DRAWGRAPHIC command 240

X

x axis, definition of 401
x-coordinate, definition of 401

Y

y axis, definition of 401

y-coordinate, definition of 401

Readers' Comments — We'd Like to Hear from You

Overlay Generation Language/370
Overlay Generation Language/370:
User's Guide and Reference

Publication No. S544-3702-04

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
INFORMATION DEVELOPMENT
IBM PRINTING SYSTEMS
DEPARTMENT H7FE BUILDING 004M
PO BOX 1900
BOULDER CO 80301-9817



Fold and Tape

Please do not staple

Fold and Tape



File Number: S370-40

Printed in U.S.A.

S544-3702-04

